£1.50

1983

THE FIRST sinclair user ANNUAL

The Complete ZX Companion

The Spectrum under the microscope

Your user problems solved

Programming in machine code

Rummaging inside the ROM

Behind the scenes at Sinclair Research

Buyers' guide to hardware, software and books

MASSES OF PROGRAMS • 1982 REVISITED

THE FIRST sinclair user ANNUAL
The Complete ZX Companion

Editor
Nigel Clark

Consultant editor
Mike Johnston

Production editor
Harold Mayes MBE

Staff writer
John Gilbert

Design
William Scolding

Editorial director
John Sterlicchi

Advertisement director
Simon Horgan

Advertisement manager
John Ross

Editorial/production assistant
Margaret Hawkins

Managing director
Terry Cartwright

Chairman
Richard Hease

Sinclair User Annual is published by ECC Publications Ltd. It is in no way connected with Sinclair Research Ltd.

Telephone
All departments
01-359 7481

If you would like to contribute to any of the Sinclair User group of publications please send programs, articles or ideas for hardware projects to:

Sinclair User and Programs
ECC Publications
30-31 Islington Green,
London N1 8BJ

Programs should be on cassette and articles should be typed. We cannot undertake to return them unless a stamped-addressed envelope is included.

We pay £10 for each program published and £50 per 1,000 words for each article used.

ISSN No. 0263-9858
Origination by
Outline Graphics.
Printed by
Eden Fisher (Southend) Ltd

Distributed by
Spotlight Magazine Distribution Ltd,
1 Benwell Road,
Holloway,
London N7
01-607 6411

CONTENTS

NEWS

During the last year there has been vast change in the world of Sinclair computers. The launching of the Spectrum at Earls Court Computer Fair in April was perhaps the major event but we find that there were many other interesting items.

Home computers 'are here to stay'

IF ANYONE had lingering doubts that home computers, and particularly ZX computers, were here to stay, the television news report in September of the Prime Minister demonstrating a Spectrum to her Japanese counterpart must have dispelled them for good. Information Technology Year, perhaps, but if it was anyone's year it was surely the year of Clive Sinclair. The year 1980 was the beginning of the ZX phenomenon and 1981 its consolidation. Now 1982 has been the year of a ZX explosion. The ZX-80 was one of the most popular computers ever but it was eclipsed by the meteoric rise of the ZX-81, sales of which by this year had topped the half-million mark. The Spectrum could do even better.

In the same period, a minor industry of independent producers has grown around Sinclair products, offering unrivalled support and expansion for the machines. An avalance of hardware and software, books and magazines has poured out, sometimes from the least-expected places the length and breadth of the country. At a time when British industry in general has languished in the grip of the worst recession since the 1930s, in this area at least business has been booming.

Sinclair Research reported a turnover of £30 million at the end of the last financial year, with record profit, attributable mostly to its computer business. In the short time since he launched the ZX-81, Clive Sinclair has become a millionaire.

It became clear more than a year ago that as a computer manufacturer Sinclair had very definitely arrived. The agreement he signed late in 1981 with the Japanese company, Matsushita, to export computers to Japan really set the seal on it. One of the most dramatic demonstrations of the popularity of the machines was the ZX Microfair in January, when a fairly modest exhibition of Sinclair software and add-ons was besieged by thousands of ZX enthusiasts who packed London's Central Hall, after waiting up to two hours to get in, and bought almost everything in sight.

Many of the 60 or so exhibitors were unable to cope with demand and found that they sold faster than they could take in additional supplies.

The continuing high demand for Sinclair-related products has led to a rapid expansion in the ZX add-on industry in the last 12 months. The few dozen firms of last year has grown to several hundred, with new companies and products appearing almost every week. Many of the firms began modestly as part-time, one-man businesses operating from a back room — some are still doing so — but have gone on to become full-time professional companies with a large turnover and exporting widely overseas.

Established companies, too, have been entering the market during the year, as the potential of the large ZX populace has become clearer. With the bigger number of suppliers, there is a much wider choice of products and users can be more discriminating.

Keyboards or additional memory, for example, are available in different quality, size and type and the buyer has the choice of up to a dozen suppliers. With software, too, there has been a marked increase in quality compared to the early days, when users were desperate to buy anything at all which would run on their micros. Some of the machine code games programs written for the ZX-81 and Spectrum can rival offerings for much more expensive machines.

Mike Johnston, organiser of ZX-Microfairs.

If Sinclair had done no more than continue to produce the ZX-81 he would have prospered but in April he surprised us all once again. After months of rumour and speculation — and not a few enigmatic comments from Sinclair himself — the Spectrum was launched, to the delight of ZX fans and the annoyance of recent ZX-81 buyers.

The machine was certainly impressive. Sinclair was at his most avuncular at the preview and he introduced the features of the machine — extended Basic, colour, sound, high-resolution, networking and discs with all the aplomb of a conjurer pulling live rabbits from a hat. The assembled journalists, who are not normally impressed by such displays, applauded. One editor went so far as to order a machine on the spot — an extremely wise move as it later proved.

Many people had expected that the ZX-81 would be superseded by the new machine. Instead, the manufacturer announced a new range of software support and a reduction in the price of the 16K RAM pack already undercut by competing products — to maintain a differential price from the Spectrum. That meant a price difference of only £25 between a 16K ZX-81 and a 16K Spectrum, which need not seem a great deal, in view of the greatly-enhanced capabilities of the new machine. That fact did not escape the public, it seems, and there were reports of a fall in ZX-81 sales. Certainly the independent producers noticed a distinct chill after April and orders began to sag alarmingly for software and hardware for the ZX-81.

Meanwhile, after an extremely promising start, the Spectrum went into a nose-dive. Customers who had ordered in April found that the promised four-week delivery

Clive Sinclair at the big event of the year

became eight or 10 weeks. Those, it proved were the lucky ones. For others, the weeks became months before they received delivery. In August the company announced a 12-week delay in delivery and was later under fire from the Advertising Standards Authority for continuing to include the 28-day delivery in its advertisements. Most of the problems were with the 48K machine, which most people were ordering.

By September only about 25,000 of an estimated 40,000 orders had been completed. Earlier Sinclair had taken the unusual step of writing to purchasers offering a £10 voucher, a free demonstration cassette, and an apology for the delay. Alternatively they could have a refund.

The company certainly had problems; apart from the huge demand for the Spectrum, there were timing problems with the original design shortly before the first orders were to be despatched. Later, there were more problems with the re-designed board and a high failure rate on the production line. In addition, the factory producing the machines was reported at one point to have closed for a three-week annual holiday.

Sadly, none of that was too great a surprise to anyone who had experienced similar delays with ZX-80, ZX-81 and ZX printers. Inevitably there are delays with production of a new and particularly a high-technology product but would it be impossible to produce a few before announcing them? It is perhaps a little unfair to single out Sinclair in this respect, despite the blood-lust engendered in some people by a three-month wait for such a desirable object. Would-be purchasers of some other systems have waited as long, or even longer.

In the late summer and autumn a large number of cheap — i.e., less than £200 — colour computers, many intended for the same market, appeared or were announced, not to mention those companies which discovered suddenly that they can sell their products for £200 less since the Spectrum appeared.

Sinclair will probably not worry too much about announcements — the usual rule of thumb is to add six months to projected dates — but the fact remains that the existing competition is hot on his heels. All of which should be good news for the consumer, who can perhaps expect an improved standard of service as a result.

The delay in Spectrum deliveries failed to deter the independents, who had already decided that the Spectrum would be a winner and proceeded to produce an impressive selection of arcade games, disassemblers, utility routines, as well as hardware add-ons — amplifiers, RAM, I/O ports — within two months of the launch.

Whatever temporary setback the ZX-81 may have had in the U.K., however, it appears to be taking the nest of the world by storm. As well as the Japanese and now the European distribution, the company signed a mutual technology agreement in March with Timex, an American firm, which manufactures the U.K. ZX machines for Sinclair.

Timex launched a ZX-81 look-alike this autumn in the States as the first fruit of the agreement. The new micro is called the Timex Sinclair 1000 and is being distributed through the company's numerous retail outlets in the U.S. It is reported to be breaking sales records.

At home, Sinclair has broken more new ground with the acceptance this summer of the Spectrum for the Government Micros in Primaries scheme. The scheme provides grants towards the purchase of selected microcomputers for use in primary schools. Sinclair Research is offering additional benefits to schools which choose the Spectrum — a ZX printer and vouchers, exchangeable for more ZX products, as well as the educational language, Logo.

If much of the foregoing is new to you, may have missed one significant event this year, which modestly we have omitted so far — the appearance in March of Britain's first exclusively ZX magazine, *Sinclair User*, and its sister publication, *Sinclair Programs*, joined lately by the magazine for the hardware enthusiast, *Sinclair Projects*. Since September, *Sinclair User* has also incorporated *Spectrum User*.

One major change noticeable during the year is the appearance of the ZX-81 in high street stores. It has been available in one major chain store for more than a year but since August — when Prism Microproducts took over part of the retail distribution — it has appeared in many more outlets.

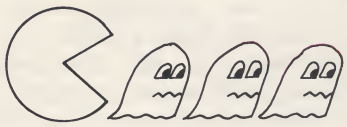
The combination of instant availability and low price should prove irresistible. Sinclair has already claimed a 500 percent increase in sales since the price was reduced, which could be very good or very bad, since we do not know what the 500 percent is on. In any event, there is considerable life left in the ZX-81 and with the range of hardware and software support available for it, it looks like being with us for the foreseeable future.

SPECTRUM COMES TO LIFE !

ABBEX

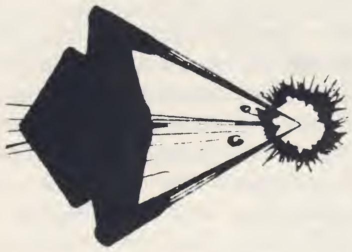
THE GAMES PEOPLE

SPOOKYMAN



EAT THE PILLS AND KEEP AWAY FROM THE GHOSTS. EAT THE VITAMINS AND CHASE THE GHOSTS. FULL MACHINE CODE ARCADE ACTION AND SOUND. DESIGNED TO MAKE YOUR ADRENALINE RUN HIGH. PRICE £4.95

COSMOS



YOU HAVE THE JOB OF PROTECTING A CONVOY FROM ALIENS AND METEOR STORMS. WATCH OUT FOR THE SPACE MINES AS YOU PLAY THIS AMAZING MACHINE CODE ARCADE ACTION SIMULATION PROGRAM. PRICE £4.95

WRITE OR PHONE FOR LIST OF GAMES FOR OTHER COMPUTERS OR JUST MORE GAMES

ABBEX ELECTRONICS LTD

20 ASHLEY COURT, GREAT NORTHWAY, LONDON NW5

Tel: 01-203 1465

PLEASE COMPLETE IN BLOCK CAPS
NAME (Mr/Mrs/Miss)..
ADDRESS..
(CARD HOLDERS)..
POST CODE........................ PHONE NO..

Make cheques payable to ABBEX
I enclose my cheque for £

I wish to pay by ☐ CHEQUE ☐ VISA ☐ ACCESS
Please debit my creditcard a/c
NUMBER ☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐

Signed..
date..

Please send me the following
(Allow 14 days for delivery)

POST BY DEC 8th and receive your order in time for Christmas

☐ **SPOOKYMAN**
PRICE £4.95

☐ **COSMOS**
PRICE £4.95

SINCLAIR USER CLUB

Having launched *Sinclair User* with our April edition, we thought the next service we should offer users of Sinclair computers was a User Club.
We began in June and it has been growing steadily since.
In this short section we meet the first member, who lives in Hull,
and provide a list of services available, as well as some of the special offers which have been available during the last six months.

ZX-81 helps at work and play

THE first member of the Sinclair User Club is an enthusiastic Sinclair supporter. He was one of the first to obtain a ZX-81, he took a year's subscription for *Sinclair User* as soon as it was announced, and was among the first to order a Spectrum.

"When Sinclair brought out the ZX-80 I decided to wait but when the ZX-81 arrived I put in an order straight away", said Peter Lown, 62, of Strathmore Avenue, Hull, North Humberside. He bought it with the 16K RAM pack and the printer.

A telecommunications manager with a large international company, Lown has made full use of his system both for work and in his hobby of motor sport.

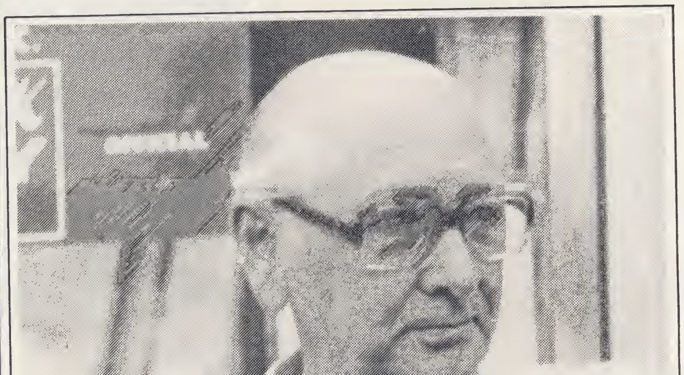
"I use it to run telephone traffic analysis. It is not complicated, just number-crunching, but there is a good deal of work involved and it saves me plenty of time", he said.

He has developed a program to display details and results of motor rallies. A friend has built him a regulator for converting 12V supply to the 9V needed for the Sinclair to be used away from a mains supply.

It had its first showing in the Cossack Rally in North Humberside. At the start it listed details of the competitors, scrolling them up so that all of them could be included. At the end it provided the results.

"It took about 25 minutes to input all the figures but we were able to do it as the cars arrived, so the results were ready within two minutes of the last competitor finishing, which was the time needed to sort through all the information," Lown said.

He first had contact with computers 25 years ago and was a founder member of the Hull and district branch of the British Computer Society. In those days he was working on a National Elliott machine which, though having the same capacity as the ZX-81, was the size of "a side of a house".



Perfect add-on for the Sinclair owner

AS A SERVICE to readers of *Sinclair User*, we set up the Sinclair User Club. For £12 a year we offer members a wide range of benefits which are not otherwise available to all users of Sinclair machines.

They include a telephone problem-answering service, a bi-monthly cassette-based newsletter and special discounts on a wide range of software and hardware add-ons.

The telephone service permits members to obtain expert advice on the use of their Sinclair machines immediately. The number is included in the cassette newsletter.

The cassettes, the first of which was sent to members in August, contain a large amount of information essential to people who wish to make full use of their ZX-81 or Spectrum. The items covered include machine code, for which a series of articles has been written, help with queries, programs and information about club members.

Each month we arrange a special Star Offer, open only to members, to allow them to take advantage of many of the useful items on the market. Discounts which we have been able to offer during the year include 10 percent off the cost of the William Stuart Systems ZX-81 Music Synthesiser and all the Thurnall Electronics range of hardware and 20 percent off the ranges of software from J K Greye and Micromega.

To join, complete the form on the next page and send it with the fee to the address shown. It is important that you state whether you have a ZX-81 or a Spectrum, so that we can send the correct cassette.

Britain

Aylesbury ZX Computer Club: Ken Knight, 22 Mount Street Aylesbury (5181 or 630867). Meetings: first Wednesday and third Thursday of the month.
Doncaster and District Micro Club: John Woods, 60 Dundas Road, Wheatley, Doncaster DN2 4DR; (0302) 29357.
Edinburgh ZX Users' Club: J Palmer (031 661 3183) or K Mitchell (031 334 8483d). Meetings: second Wednesday of the month at Claremont Hotel.
EZUG-Educational ZX-80/81 Users' Group: Eric Deeson, Highgate School, Birmingham B12 9DS.
Furness Computer Club: R J C Wade, 67 Sands Road, Ulverston, Cumbria (Ulverton 55068). Meets every other week on Wednesday evenings.
Glasgow ZX-80/81 Users' Club: Ian Watt, 107 Greenwood Road, Clarkston, Glasgow G76 7LW (041 638 1241). Meetings: second and fourth Monday of each month.
Hassocks ZX Micro User Club, Sussex: Paul King (Hassocks 4530).
Inverclyde ZX-81 Users' Club: Robert Watt, 9 St. John's Road, Gourock, Renfrewshire, PA19 1PL (Gourock 39967). Meetings: Every other week on Monday at Greenock Society of the Deaf, Kelly Street, Greenock.
Keighley Computer Club: Colin Price, Redholt, Ingrow, Keighley (603133).
Merseyside Co-op ZX Users' Group: Keith Driscoll, 53 Melville Road, Bootle, Merseyside L20 6NE; 051-922 3163.
National ZX-80 and ZX81 Users' Club: 44-46 Earls Court Road, London W8 6EJ.
North Hertfordshire Home Computer Club: R Crutchfield, 2 Durham Road, Stevenage; Meetings: first Friday of the month at the Settlement, Nevells Road, Letchworth.
North London Hobby Computer Club: ZX users' group meets at North London Polytechnic, Holloway Road, London N7 each Monday, 6pm.
Nottingham Microcomputer Club: ZX-80/81 users' group, G E Basford, 9 Holme Close, The Pastures, Woodborough, Nottingham.
Orpington Computer Club: Roger Pyatt, 23 Arundel Drive, Orpington, Kent, (Orpington 20281).
Perth and District Amateur Computer Society: Alastair MacPherson, 154 Oakbank Road, Perth PH1 1HA (29633). Meetings: third Tuesday of each month at Hunters Lodge Motel, Bankfoot.
Scunthorpe ZX Club: C P Hazelton, 26 Rilestone Place, Bottesford, Scunthorpe; (0724 63466).
Sheffield: Andrew Moore, 1 Ketton Avenue, Sheffield S8 8PA would like people interested in starting a club in the area to contact him enclosing a stamped-addressed envelope for details.
Sittingbourne: Anurag Vidyarthi (0795 73149). Would be interested to hear from anyone who wants to start a club near the Medway towns.
Swindon ZX Computer Club: Andrew Bartlett, 47 Grosvenor Road, Swindon, Wilts SN1 4LT; (0793) 3077. Monthly meetings and software library.
Thames Valley ZX Users' Club: Richard Shepherd, 22 Green Leys, Maidenhead, Berkshire SL6 7EZ; (0628) 21107 (evenings and weekends). Hopes to start meetings on a regular basis.
Worle Computer Club: S W Rabone, 18 Castle Road, Worle, Weston-super-Mare BS22 9JW (Weston-super-Mare 513068). Meetings: Woodsprings Inn, Worle, on alternate Mondays.
ZX-Aid: Conrad Roe, 25 Cherry Tree Avenue, Walsall WS5 4LH (Walsall 25465). Please include sae. Meetings twice monthly.
ZX Guaranteed: G A Bobker, 29 Chadderton Drive, Unsworth, Bury, Lancashire. Exchanges information and programs throughout the country.
ZX-80/ZX81 Users' Club: PO Box 159, Kingston-on-Thames. A postal club.

Overseas

Belgium, France, Luxembourg: Club Sinclair, Raymond Betz, 38 Chemin du Moulin 38, B-1328 Ohain, Belgium (322 6537468)
Belgium, Netherlands: Microcomputer Vereniging BZW, Paul Glenisson, Priester de l'Epéestraat 14, B-1200 Brussels, Belgium (322 7349954)
Denmark: Danmarks National ZX-80/81 Klub (DNZK), Jens Larson, Skovmosevej 6.4200 Slagelese, post giro 1 46 24 66.
ZX-Brugergruppen i Danmark, Boks 44, 2650 Hvidovre. Gratis medlemskab og gratis blad til enhver interesseret.
East Netherlands: Jonathan Meyer, Van Spaen Straat 22,6524 H.N. Nijmegen; (080 223411).
Germany: ZX-80 Club, a postal club; contact Thomas Jenczyk. Hameln, Postfach 65 D-3250 Hameln, Germany.
Indonesia: Jakarta ZX-80/81 Users' Club, J S Wijaya, Jakarta, Indonesia.
Republic of Ireland: Irish ZX-80/81 Users' Club, 73 Cnoc Crionain, Baile Atha, Cliath 1.
Singapore: Sinclair Users' Group: Eric Mortimer, 1D Wilmer Court, Leonie Hill Road, Singapore.
South Africa: Johannesburg ZX80/81 Computer Users' Club: S Lucas, c/o Hoechst SA (Pty) Ltd, PO Box 8692, Johannesburg. Teaches Basic and machine code, interest in hardware.
Johannesburg ZX Users' Club: Lënnërt ER Fisher, PO Box 61446, Marshallstown, Johannesburg.
Spain: Club Nacional de Usuarios del ZX-81, Joseph-Oriol Tomas, Avda. de Madrid, No 203 207, 10, 3a esc. A Barcelona-14 Espana.
International ZX Spectrum Club: Gabriel Indalecio Cano, Sardana, 4 atrico 2a, San Andres de la Barca, Barcelona. Send international reply coupon. Produces a bi-monthly magazine.
United States: Bay Area ZX-80 User Group, 2660 Las Aromas, Oakland CA 94661. — Harvard Group, Bolton Road, Harvard MA 01451; (617 456 3967).

MEMBERSHIP FORM

I wish to join the Sinclair User Club and enclose my subscription of £12

Name...

Address ....

....

Send your coupons to Sinclair User Club, ECC Publications, 30-31 Islington Green, London N1 8BJ. Cheques should be made payable to Sinclair User Club.

SOFTWARE SCENE

Buying a Sinclair computer is only the start of spending by many people. The major expenditure is in software. When the ZX-81 was launched there was little available but that has changed dramatically during the last year, with cassettes available for games of all kinds and more serious uses such as education and business. Our writers assess the quantity and quality of what is on the market.

Overview by John Gilbert

Games are still dominating

THE GROWTH of the ZX software market started slightly less than a year ago. Since then it has branched into three areas — games, business and educational. At the moment the market is centred firmly on games programs which, for the ZX-81 at least, are becoming very sophisticated. Initially that was not so, as firms were taking their first feeble steps into the software market and were discovering what the ZX-81 could do.

Some of the games cassettes produced by Sinclair for the ZX-81, called Super Programs, are examples of the poor quality of some of the earlier programs. They were produced by International Computers Ltd and the screen displays look as if they were from one of the company's remote terminal printers and not a durable medium like a television screen, with thousands of display and movement possibilities. The 16K Super Programs were only slightly better than 1K programs. What is surprising is that they were vetted by Sinclair Research. ICL has just introduced a range of software for the Spectrum. Those programs, including the inevitable games pack and Fun to Learn programs, look like ZX-80 and ZX-81 programs with colour and sound and a few high-resolution graphics characters added to make the screen look more interesting. The programs were written for the 16K Spectrum and demonstrate none of its real capabilities.

The leaders in the software market seem to be those who produce well-presented games, with packing just as important as game content. Companies like Qucksilva and Bug-Byte produce colourful cassette case inserts and that is one reason why their programs sell well. Customers buy programs on first impressions, which usually means the description of the game on the cover and the picture are the decisive factors. Quicksilva Asteroids and Invaders packaging are colourful and help sell the product.

While there are many games which are not value for money, several good programs have stood out from the rest during the year. Those programs have developed new ideas and given a twist to old ones. For instance, three-dimensional displays have caused a stir, in particular the J K Greye 3D Monster Maze and 3D Defender. There are other 3D mazes but the Greye version seems to be the best.

Another unwelcome trend in the early development of the software market was the conversion to ZX Basic of games which had been played on mainframe computers. One such game is Star Trek. Usually it was played via a remote teletype terminal which provided very limited display potential. That limited graphics format seems to have been transferred to the ZX versions of the game.

Several companies, including Silversoft, have produced Star Trek games with close resemblance to the mainframe version. The video display is, to say the least, drab. When the game was converted to microcomputer format the screen display could have been enhanced as well, making more use of moving graphics.

Versions have been produced for the Spectrum which are slightly better, with colour and sound, but not much better. There is nothing wrong with reproducing a game for various micros — unless it happens to belong to somebody else — but it is unnecessary to retain the rigid display of mainframes.

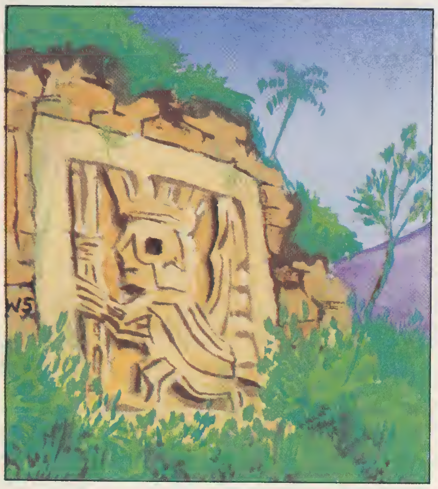
Discussions about software piracy and copyright have featured in the news recently. Until the various points of law about software copyright are settled, the situations where companies are at variance will recur again and again.

The companies which produce ZX software have so far had few problems.

Business software takes an ever-expanding share of the market, although it is still a small portion. Companies producing business software for Sinclair computers usually have made a mark on the software market with programs for other machines. The programs may not be as capable as programs for such machines as the Apple, which uses discs, but companies like Hilderbay have done a good job in taking professional business programs to the ZX market.

With the release of the Spectrum, the RS232 interface and micro drives, the ZX software market should attract many more business users. Educational software is something of a neglected area in the ZX market. Some companies like ICL, Calpac and SciSoft, have made it their business to produce educational programs. The quality so far is good but the range seems to embrace only pre-school and junior education.

A computer, especially one as cheap as the ZX-81 or Spectrum, is ideal as a teaching aid. It does not become angry with a child for not learning; it will allow tests and information to be accessed when needed; and it will mark tests done by a child speedily. It also seems



that children have a higher retention rate when learning with a computer. There is no reason why the software range should not include programs for A level candidates or even university students.

It would be disappointing if an area in which ZX computers are so useful is not exploited more widely and now that the Government grant scheme has been extended to ZX computers, there is no reason why it should not be.

Machine code and advanced Basic programming is becoming of interest to Sinclair users. The number of assemblers, disassemblers and toolkits on the market is helping users to explore their machines and exploit them to an even greater degree.

The use of an assembler makes it possible to forget all about machine code as numbers and to learn the Z-80 operation codes. The idea is good but many of the assemblers produced for ZX machines are difficult to use. The essence of success when dealing with other than a high-level language is user friendliness. If a program is to be helpful to a user who is breaking new ground that must be the prime consideration. Many of the earlier programs were very short of explanatory material and introduction booklets were almost unheard of. The most companies used to include with a tape was a four-page leaflet with short operating instructions.

Fortunately it is one area of the market where there has been definite improvement. Companies such as Artic and Picturesque provide handbooks with their assemblers and monitors. That makes comprehension much easier and gives the user so much more power over the computer.

For the user who does not want to use machine code, many companies have produced a variety of toolkits. They are machine code routines which will enhance program graphics, re-number programs, and find the amount of user memory remaining in the machine. JRS Software produced a Toolkit to help the Basic programmer to type-in the program code and the Graphics Toolkit can be added to enhance the graphics with drawing routines and scroll left and right.

The software industry will change further in the next few months with the introduction of Micronet, a system which, among other things, will allow users to receive programs for their machines from a central database via the telephone. Eventually that could mean the phasing-out of software distribution as it now exists. Cassettes could disappear from over-the-counter sales, as programs will be down-loaded directly into a machine and stored on disc or tape.

Telesoftware is workable and there will be more prolific use in the near future. Until then, cassette is the sales medium for ZX software.

The market has expanded well in the last year and should continue to grow. Games software has been the most-explored area so far but soon, with the expansion of the Spectrum, business and educational uses will expand. It is the hardware which inflicts limitations on a computer but it will be the software which will show whether ZX computers can be used for serious applications.

Adventure games

Ripping yarns

IF YOU enjoy delving into rat-infested dungeons, crawling through dimly-lit caves, and wandering round long-abandoned mine workings, Adventure is for you. Your powers of memory, logic and patience are tested to the limit as you take life-and-death decisions in your search for fame and riches — all without leaving the comfort of your chair and with only one risk to your physical well-being. That is a severe headache after bashing your head against the wall when your latest seven-minute Adventure failed to load for the fifth time.

Adventure is a role-playing game, where after being told about your immediate environment, you can enter commands such as "Go west", "Take keys", "Throw knife". The computer will then provide an appropriate response, such as a new room description, "You missed", or some other message. You may encounter fabulous treasures, or perhaps only mundane objects, although they may, if used correctly, be the key to a still greater fortune.

You will probably need to make a map as you proceed to have some chance of re-tracing your steps to the entrance. Natural — and unnatural — hazards abound to prevent your escape and it should take many delvings to discover all that an Adventure contains.

Computer Adventures are a surprisingly recent invention, the original having been written on a mainframe in 1976 by William Crowther and Don Woods at Stanford University in the U.S. It is a remarkable achievement when you consider that, until recently, Fortran, the language they used, had none of the string-handling facilities of Basic. Until two years ago, Adventure was strictly the preserve of computer professionals but the growth of micros has resulted in far wider use.

For this review of Adventure games for the ZX-81, we looked at every Adventure and Adventure-

type game which could be. They vary a great deal, from the traditional descriptive type to real-time graphics games, so we have tried to judge each on its merits rather than compare one to another.

One employer told his staff recently to take Adventure off the computer at work, as they were running out of disc space. That being such an unreasonable instruction, they checked how much memory it used and found that the program and the data it required occupied more than a quarter of a million bytes. So, not surprisingly, all the ZX-81 games require 16K RAM packs.

The logical program with which to start is the Abersoft Adventure, which attempts openly to pack as much of the Crowther and Woods' original into 16K as possible. As the writer was usually hopelessly lost in the maze in the Adventure at work, he was hoping that perhaps lack of memory had forced the omission of the maze from the Sinclair version. Not a chance; on the first attempt he went straight into it like a homing pigeon. The maze seems like quicksand; the more you struggle to escape the deeper you sink.

At that point he asked a friend who had acquired a good knowledge of the original Adventure to try from the beginning. It was found that all the locations and objects were in the correct places, although the location descriptions were shorter. It was even discovered there was a new area not entered previously.

The program is written in 13K of machine code and 1½K of Basic, with more than 70 words which can be recognised as commands or objects, and a large number of locations. The game has been written with remarkable efficiency; location descriptions are built-up from individual words and phrases rather than stored *en bloc*. Being machine code, the program is very fast. At any time you can find your score and also save your current position on tape to continue later.

At £10, it is an expensive program but it is a remarkably good version of the original Adventure and well worth the money if you want to see what sparked the entire process.

Hilderbay is best-known for being the "serious" software company supplying business programs, but it has also forayed into the world of games and its Gold is a non-graphics 13K Adventure written in Basic. The object is to search for gold hidden in a network of caves and mines in the Yukon. All instructions are entered as a single letter, O for Open, G for Get, including references to objects such as Gold and Keys.

There is a sizeable network of logically-connected caves and rooms, although there are few objects. If you manage to find the gold, things change mysteriously. At any time you can learn your score and there is an added bonus if you manage to escape with the treasure.

At £8, with a word game included, Gold is reasonable value but could have been better with a few more objects — and word rather than letter input.

Bug-Byte offers two adventure-type games, the latest of which, Dictator, is a brilliant simulation of a banana republic; since it has no locations, no objects, and no movement of any kind, it is not an Adventure so it is not included.

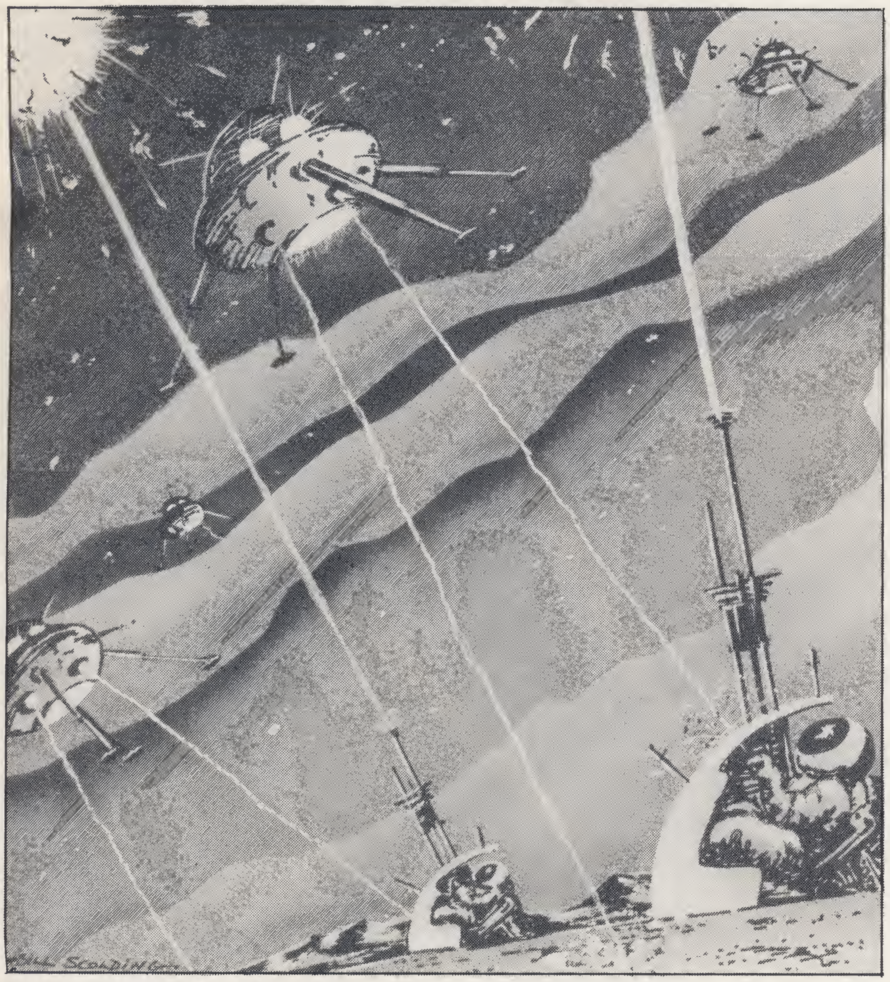
The other is The Damsel and The Beast which is a graphics-only 9K Basic game. The setting is an unlit palace of 35 rooms, containing a beast for bashing, a damsel for rescuing, and a few holes into which you can fall. That calamity should be avoided by using the limited supply of torches available and you are also provided with some powerful clubs if you should wish to knock down a wall or kill a monster.

After being set-up randomly, the layout of the rooms does not change, so a logical approach is rewarded. Movement, and the use of clubs and torches, is by single-letter commands and the beast continues to move of its own accord while you decide what to do.

The game is not desperately fast and while in progress there is little to look at, although you are provided with a map of the events at the end. There is no score given but there are three levels of play and even the easiest of them is difficult enough. It is not a game for Space Invader addicts but if logic and patience are your virtues, it is one to consider. It is on the expensive side at £6.50.

Quest, from Serious Software, is clearly based on the non-computer Dungeons and Dragons fantasy role-playing game, with the computer-man pitting his strength, constitution and dexterity against wandering monsters. It is a non-graphics game written in 12½K of Basic and is made up essentially of vampires, rats, werewolves and suchlike. If you can keep the fiends off your back for long enough there are potions, keys and other objects to be found and substantial amounts of treasure to be acquired.

Movement and fighting is done by single-key input, which can be confusing, since the same key may mean different things depending on what is happening around you, e.g., F may mean move Forward or Fire magic arrow. You can always ask for a status report and, if you can put



together sufficient treasure, the delights of different levels are offered.

At £5.95 for Quest plus a Star Trek, Mastermind and Reversi included, it is good value for the nimble-fingered.

Adventure from Simpson Software has its origins in the Crowther and Woods' original but is set in a mythical castle containing evidence of an extraordinary mixture of living beings—hobbits, dwarfs and pirates, among others. It is a non-graphics adventure with 25 logically-connected locations written in 11½K Basic.

The method of processing instructions entered by the player is unusual; instead of checking for individual words, as most Adventures do, the input string is compared in a large series of if-then statements. That makes it reasonably fast, but means there is a very limited vocabulary; e.g., you must refer to a Ming vase, not just a Vase. One weak point—answers to Yes or No questions are not validated, so a NEWLINE stops the program.

There are plenty of treasures dotted about, a hidden lower level and a maze into which you can fall. A score is kept, though our earliest attempts ended in negative scores. It is an unpretentious program, reflected in its price of £3. It would make a good introduction to anyone new to the concept of Adventure who wants to start with something reasonably simple.

Phipps Associates offers three complete Adventures on one tape, plus a detailed instruction program. All the games are written in 15K of Basic and the first two are based on Trevor Toms' Adventure-writing program from his book *The ZX-81 Pocket Book.*

Greedy Gulch is set in a Wild West ghost town with more than 20 locations, plenty of objects which have to be collected and used in a logical sequence, and a vocabulary of more than 50 words. It is not particularly fast — around 10 to 12 seconds to process instructions — but it has some graphics in the form of a simple but useful map.There is no score but to compensate there is a hint feature.

The second is the non-graphics Pharaoh's Tomb, which has more than 60 locations, made possible by giving only short descriptions, and more than 20 objects. It runs slightly faster than Greedy Gulch, despite an even larger vocabularly, and has a score but no hint feature.

It shares with the first program some poor spelling, although Phipps Associates is by no means the only culprit where that is concerned.

Magic Mountain has witches, wizards and spells, and an assortment of mystified objects. It also has a maze which, as usual, we discovered very quickly. It needed a solid guilt-ridden hour of cheating to get out, despite using one of the hints available in the form of cryptic crossword-type clues. All three programs are excellent Adventures and at £5.95 complete they are undoubtedly the best value for money of all the Adventures we have seen so far.

Artic Computing advertises three Adventures, with the rather uninspired titles of A, B and C. They are written in machine code and seem to be from the same original master program, so we can probably expect more adventures in the future.

Adventure A appears to be 12½K long but on closer inspection a fair amount of this seems to be empty. It has about 20 locations, a similar number of objects, and along with the other Artic games, a large vocabulary of more than 100 words.

The setting is an alien planet which you are trying to leave, and there is a green man to deal with, a spaceship to find, and even a computer — they are everywhere. Unlike the other two games, you cannot save your present position to return later.

Adventure B is set in an Inca temple, is 11K long and is the only one of the three to give you a score. For what it is worth, ours never went above zero. This game has 50 locations with short descriptions and more than 25 objects, not including the treasures, which, as in all the Artic games, need to be used at the proper time and in the proper combination to be useful.

There were some difficulties with this game. It was sometimes very strict about the word required at a certain point; for example, you cannot go "Up" the stairs, they must be "Climbed".

The 13K Adventure C is the biggest of the three and is set on an alien spaceship. The object is to press a control button somewhere which will release your ship from

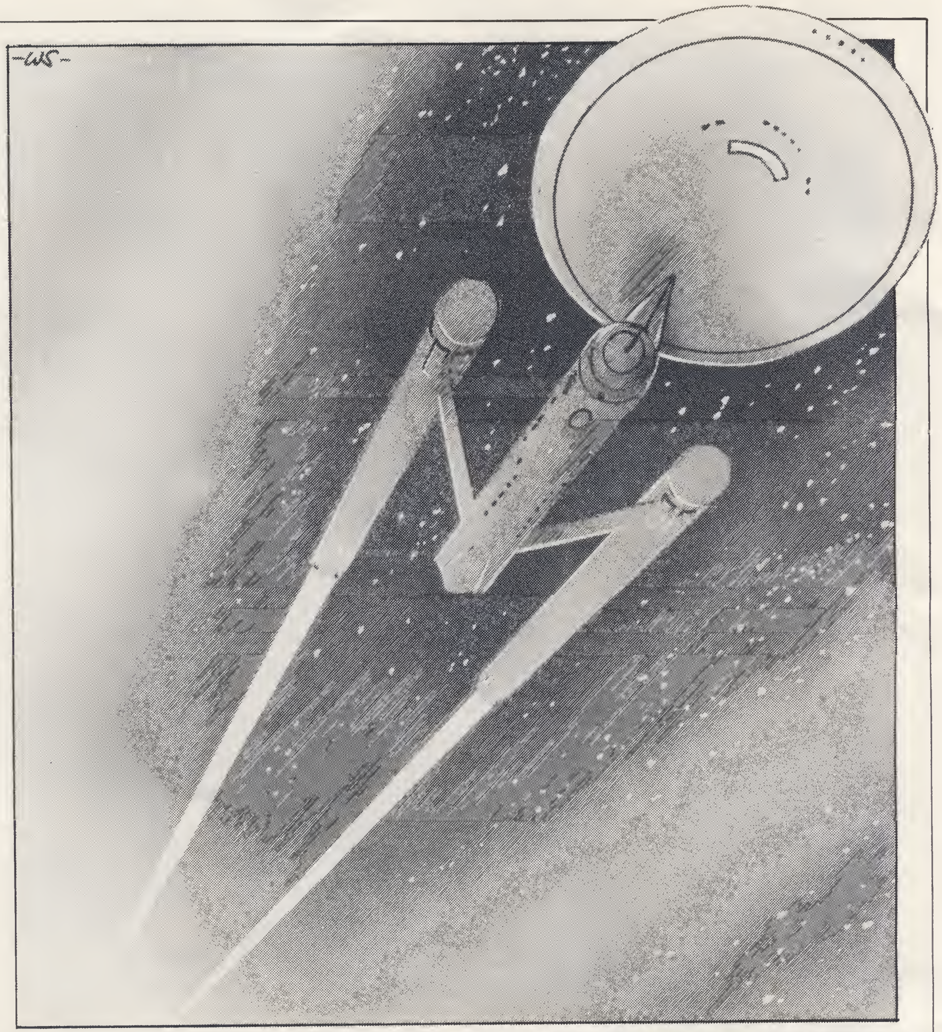
the fiendish Gravitron Beam and allow you to escape. The program contains more than 35 locations and 40 objects, and is on two levels separated by a hidden door. Despite having spent hours exploring and manipulating objects on the first level, we still have not been able to break through.

Having cheated furiously we know that, apart from the control button, the other level contains more rooms and objects and a distinctly X-rated Android we would like to meet.

All three programs respond to "Help", although rarely helpfully, and "R" repeats the room description.

Despite the large vocabulary, the response time is, to all intents and purposes, instantaneous, which makes a difficult and frustrating adventure easier to bear.

All the programs use the Artic keyboard scanning routine, which means that there is no reponse to the break key. The only way we have found to stop the programs, so that we could make a security copy, is by entering three or four Newlines and then a complete line of letters which overloads the display file and stops the program with a "5" error.



Having done that, we discovered that the instructions for Adventure C had the name of the program wrong. The filename is ADVENT C, not ADVENT as stated.

At £5, £7 and £7 for A, B and C respectively, they are all good value and will take many, many hours to master.

Catacombs from J K Greye is an all-graphics-real-time game. There is no chance of having to think about where to go next on this one, as your strength steadily drops whether or not you are doing anything.

As you move around using the standard cursor controls, the surrounding area is revealed. Each level of the catacombs comprises a random set of inter-connected rooms containing random amounts of food, F, gold, £, and monsters — O for Ore, D for Dragon. Depending on your strength you can either fight the monsters or run away and, if necessary, you can even tunnel through the walls.

The program is written in 9½K Basic and 2K of machine code. Despite the machine code, the game takes more than two minutes to set up. Something else to watch for is the Exit, X. If you go through it you have a two-minute wait for the next level to be set up.

This is a nicely-done graphics game with your strength and score, the amount of gold you have amassed, shown on-screen. At £5.95 it is a little expensive and would be greatly improved if the setting-up could be converted to machine code, since beginners may find the setting-up lasts longer than the game.

We have to admit that the Giltrole Nasty Mountain nearly had us beaten. After playing the game, studying the listing, and cheating furiously, we finally managed to get out with a score rated as "awful".

The idea is to cross a mountain via a set of seven logically-connected caves. Your tortuous path from one cave to the next is shown graphically, and the caves may contain objects, mainly edible, such as apples and carrots. The nasties are not all that fearsome, being rabbits and chickens, but they have to be treated in the proper way if you want to get anywhere.

The program is written in 12K of Basic and runs at a gentle pace. Movement and picking-up objects can be done with whole words or abbreviations if preferred but you are told your score only if you manage to get out. You can enter "Help" if you get stuck but all that happens is that the program determines whether or not it is still possible for you to escape, which is scarcely helpful.

This is a well-presented logical Adventure and £4.95 is a fair price.

Philip Joy's non-graphics Cathedral Adventure is written in 15K of Basic and describes more parts of a cathedral than we ever knew existed — more than 30, in fact. Short descriptions are given, sometimes including a cryptic clue — no pun intended — and more than 70 words are recognised, although the input processing routine can be slow, sometimes nearly 30 seconds.

Some of the treasures which are scattered around may be required later in the adventure, although we have not yet got beyond the Mad Monk to find out.

Plenty of invention has been used in working-up the locations, and some of the spelling, too, in this game, which costs £7.50.

Psion offers a tape with two sci-fi adventure-style games, written in 9K and 14K of Basic. The task facing the intrepid adventurer in Perilous Swamp is to rescue a princess and return safely, having fought, or bribed, monsters at every turn. You are given a map to help you and a new layout is produced for each game.

The monsters, their strength, and the amount of treasure they are guarding are generated randomly at each step; you have to decide how much strength to use in overcoming them, or how much to offer as a bribe.

The second Psion program, Sorcerers Island, is a cross between the first and more traditional Adventures. The detailed map is the same for each game and takes nearly a minute to display. There is a small vocabulary, move, fight, and so on entered as single letters, some objects, and even a rather ponderous maze. As you try to find the way off the island you use up your Life Points and hope to increase your Treasure Points.

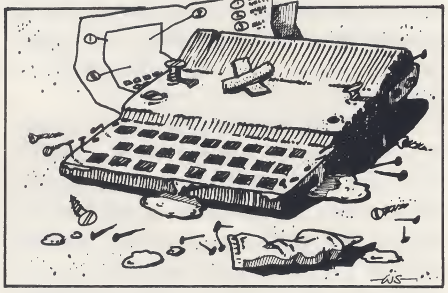
All the foregoing adventures are designed for the ZX-81. As yet, few have been converted for the Spectrum. One company which has already moved into the market is Avon Software.

It has produced a game in the classic style for adventurers. Zolan's Tomb is in two parts. The introduction and instructions are loaded first and the game can be loaded after they have been understood. The player has to search for the legendary Zolan's Tomb and the game seems to require an intelligence of near genius level to be solved.

Abersoft, 7, Maesafallen, Bow Street, Dyfed SY24 5BA.
Artic Computing, 396, James Reckitt Avenue, Hull HU8 0JA.
Bug-Byte, 98-100, The Albany, Old Hall Street, Liverpool L3 9EP.
Giltrole, PO Box 50, Rugby, Warwickshire CV21 4DH.
J K Greye, 16 Park Street, Bath, Avon BA1 2TE.
Hilderbay Ltd, 8-10, Parkway, Regents Park, London NW1 7A.
Philip Joy, 130, Rushgreen Road, Romford, Essex.
Phipps Associates, 99, East Street, Epsom, Surrey KT17 1EA.
Psion Computers, 20, Clifton Court, Maida Vale, London NW8 8HT.
Serious Software, 7, Woodside Road, Bickley, Kent BR1 2ES.
Simpson Software, 21, Tuttles Lane West, Wymondham, Norfolk.

Phil Garrett assess toolkits

Adding polish to rough programs



THERE ARE two types of programming aid and which type you use will depend on whether you want to write in Basic or machine code. Toolkit routines provide a series of machine code programs to execute such operations as line re-numbering, scroll left and right on the screen, and to indicate to the user how much memory remains. They provide extentions to the Basic language. Assemblers, dissassemblers and machine code editors are examples of the other types of programming aid. They are designed to help people programming in machine code who do not want the complications of programming in hexadecimal or looking for bugs in lists of numbers.

Even a program which is a jumble of embedded subroutines and GOTOs every other statement looks well-planned when all the line numbers increase in tens. Re-numbering is probably the main reason for the purchase of a toolkit program and is the only function included in every one at which we looked. The range of other functions offered is very wide and some of them seem to have been put in just to make a program appear more substantial, rather than because they are useful.

Hewson Consultants offers two programs, a straightforward Renumber and a more sophisticated Toolkit. Both can be used on ZX-81s with either the standard 16K RAM or with the increasingly-popular 48K and 64K RAM packs. They require the user to re-set RAMTOP before the program is loaded, which can be annoying, especially when you forget to do it.

Renumber takes 1K and a USR call re-numbers your Basic program in steps of five, starting from 5. Other values for the step size and starting number can be POKEd in and it will always re-number to the end of the program. The routine takes about 50 seconds in Fast mode to re-number 5K of Basic.

A problem with which all the re-numbering programs have to cope somehow is the very useful but very non-standard Sinclair GOTO — and GOSUB. The GOTO 1000 + 100* A and GOSUB X are seen in few, if any, other dialects of Basic, and are too sophisticated for a re-numbering program to cope with, so have to be tinkered with afterwards.

Another non-standard feature is the jump to a non-existent line number; Sinclair Basic simply continues looking until it finds a line, whereas most Basics will stop with an error if the line does not exist. A measure of the usefulness of a re-numbering program is what assistance it gives to the programmer in resolving those difficulties.

To return to the Hewson Renumber, computed GOTOs and jumps to non-existent lines are highlighted in reverse video after re-numbering. The arithmetic expressions — e.g., GOSUB 1000 + INT (10*RND + 1) — can appear rather distorted, so it would be as well to have a copy of the original program to which to refer.

Hewson's Programmers' Toolkit requires 3K above RAMTOP and includes routines to copy or delete blocks of Basic lines and a number of machine code monitor functions. The re-number works in much the same way as the previous program, except that an end number can also be given; the numbers are prompted for, rather than having to be POKEd in. The Toolkit seems to consist mainly of lines of Basic starting from 9000, which are added to your Basic program with a USR call. RUN 9000 will then run the Toolkit, which displays a menu of functions. There is a

hexadecimal loader and lister and a routine to copy bytes from one place to another.

More useful are the functions which can display the line numbers of lines containing a particular string of characters or tokens you want to find — e.g., computed GOTOs — and REPLACE, which allows a string of characters to be exchanged for another of equal length. Both routines are slow to execute.

The program is not very robust; the hex lister did not work, the re-number duplicated line numbers when the increment was set too large, the program crashed several times. The instructions for both programs are brief, but adequate. Renumber costs £4.95 and Programmers' Toolkit £6.50.

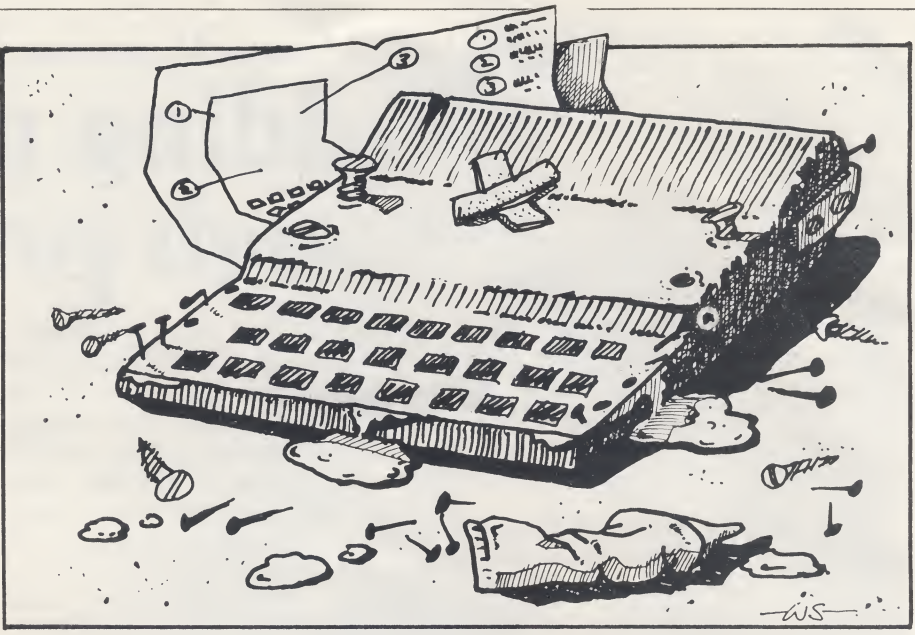
The JRS Software Toolkit also requires the user to re-set RAMTOP and takes 1K. A great deal has been packed into that 1K but ease of use seems to have fallen by the wayside. The re-number requires all statements such as GOTO 25 to be changed to GOTO 0025 before it will work. Computed GOTOs and jumps to non-existent lines are ignored completely. Starting line and increment can be changed with POKEs and the routine needed eight seconds to re-number 5K.

There is a search-and-list function which could be used to find all occurrences of GOTOs and GOSUBs, so that they could be changed to the required format. There are also search and replace and memory-left routines, plus three graphics routines, Hypergraphics mode alters the start address of the ZX-81 ROM character table and produces interesting but useless effects; Fill fills a specified number of lines with a chosen character; and Reverse inverts as many lines on the screen as required.

At £4.95 the program demands a great deal of care and effort from the user, which surely is not the purpose for which utility programs are intended.

The ACS Software Progstore allows a small — fewer than 2,750 bytes — Basic program to be stored above RAMTOP. The program can then be called, with USR, and acts rather like a subroutine. Any variables used in the stored program must already exist.

On the other side of the tape are



four small Toolkit programs which can be used with Progstore. Hexloader and Hexlister are obviously fillers and a waste of time; no addresses are shown, so it is difficult to interpret what appears on the screen. Progmod-1 allows larger programs to be stored above RAMTOP by modifying Progstore. Renumber works only in steps of 10 from line 10 and, incredibly, the entire program will crash — if it encounters a computed GOTO — or jump to a non-existent line number.

The other ACS cassette contains Progmerge, which allows all or part of a Basic program to be stored above RAMTOP and then merged with a second program, the whole lot then being re-numbered. The instructions did not indicate what size of program could be stored and a 3K program was too big.

The ACS assembler and disassembler programs are impressive, so the poor quality of its re-number routine is surprising. It can be used only with the utmost care. The instructions are clear, with helpful examples. Progstore/Toolkit costs £7.50, and Progmerge £5,50.

ACS, however, has now produced Progmerge (version two) which is said to have overcome the problems. The program costs £5.50 and anyone with the first version who wants it updating can have it done by ACS at a cost of £1 plus 25 pence for postage and packing.

At 4K, the dK'tronics ZXED toolkit is bigger than the others and re-sets RAMTOP automatically. The program is controlled from six lines of Basic 9990-9996 which are added to your program with a USR call; RUN 9990 starts the toolkit and gives an inverse E prompt, waiting for one of 11 commands. Whichever is chosen, full prompts are given and, if anything goes wrong, 10 special error codes will identify why and where it happened.

Renumber prompts for start and end line numbers, new base and increment. It then looks through the Basic and if it encounters a computed GOTO it will stop with report "Q/line no.". The user can then insert a REM into the offending line and continue with the re-numbering. Impressively, jumps to non-existent line numbers will be re-numbered correctly, e.g., 5 REM 15 REM 25 GOTO 10 will become 10 REM 20 REM 30 GOTO 20. It took less than one second to re-number 5K of Basic.

Find will display the lines — not just the line numbers — containing a specified string, so it can be used to find the REM GOTO X lines after re-numbering, and Alter will replace a string. Blocks of lines can be copied, moved or deleted, stored above RAMTOP and re-inserted into another program. Bytes tells you the amount of free memory remaining.

It is a most impressive program, fast in execution, with clear and full instructions and helpful error codes. At £6.95 for the cassette version and £9.95 for an EPROM version, it stands out from the rest of the field.

Judging by the popularity of Space Invader-type arcade games for the ZX-81, and of books such as

Toni Baker's *Mastering machine code on your ZX-81*, it seems that ZX-81 owners want not only to run machine code programs but write them, too.

Bug-Bytes ZXAS assembler is in 5K of machine code, with a few lines of Basic to operate it. The program re-sets RAMTOP automatically and loads itself above it, so that assembler source programs can be loaded and saved separately. Lines of mnemonics are entered in REM statements, with multiple instructions allowed, provided they are separated by semi-colons. Up to 256 labels can be used in the form :LO to :L255, and comments may be placed after a "*". Full-stops are used instead of commas — e.g., LD A.H — which makes typing instructions easier and numbers may be entered in decimal or hex. When the assembler is run, you are prompted for the starting address for the resulting machine code.

It is worth using a REM statement at the start of the program and compiling the machine code from 16514. The assembler code is then displayed on the screen in the format source line number; address (in hex); opcode and data (in hex); Z-80 mnemonic. If there is an error, the assembler stops with an error code, so it is not difficult to build a syntactically-correct source program.

There is at least one bug in ZXAS; the SUB A,n instruction does not work but it can be replaced by AND A; SBC A,n which does the same. ZXAS is a remarkable program and is excellent value at £5.

The other ZX-81 assemblers are produced by ACS Software and Artic Computing. The ACS one is similar to ZXAS in size and operation, with instructions entered in REM statements and labels available in the form Q.1:Q.255:.

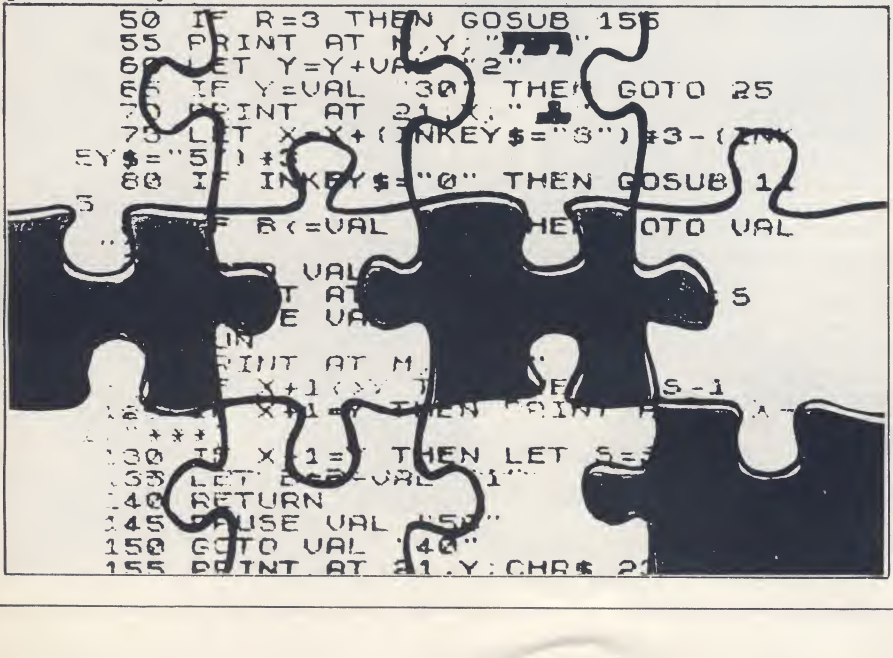
Data must be entered in decimal rather than hex and there is a useful DFB function which allows you to specify the contents of a particular byte during assembly, so you can have messages embedded in your machine code.

The assembled listing display is slightly different from ZXAS; you are given the decimal address, hex op-code and data, and then the mnemonic.

The ACS assembler is also excellent value at £5.50 and the use of either this program or ZXAS is the single biggest step to proficiency in machine code programming.

Neither program sets out to teach assembler, so a book will also be needed. The thorough but expensive *Programming the Z-80* by Rodney Zaks is recommended but there are now several books available specifically for machine code programming on the ZX-81.

The ZX Assembler from Artic Computing provides an easy introduction to writing machine code routines for the ZX-81. The package includes a glossy manual which takes the beginner through the various operations of the program and includes a reading list of recommended books. The program compiles the Z-80 op-codes of the user program from REM statements. Ordinary text which is to be printed on the screen can be written directly into the code to be assembled and will be turned automatically into hexadecimal. Disassemblers convert machine code into mnemonics, making it easier to analyse and amend. The ACS disassembler can be used at the same time as its assembler and provides mnemonic listings in the same format. All addresses are shown in decimal and destination addresses are shown for relative jumps rather than the displacement, which is an excellent idea.



All the other disassemblers have additional features to assist with editing and debugging machine code. Campbell Systems 4K disassembler uses plenty of Basic and its machine code occupies the 16514 onwards area of RAM. It has a handy facility to step backwards and displays contents in hex, with addresses and mnemonics in decimal.

You cannot dump direct to the printer and you have to use the break key to return to Basic. Machine code can be entered and individual bytes changed, using hex. This program, price £4, has been available since June, 1981 and has perhaps been superseded by some of the others on the market.

The Aylesbury ZX Computer Club has decided courageously to enter the software fray with its disassembler. It is a very large program (14K) and runs a little slower than the others. The display, which can go to screen, printer, or both, is unusual; addresses, contents and mnemonic data are given in both hex and decimal and the display allows one line for each byte. There is also a facility to enter machine code from address 30000 in either hex or decimal or an Edit function to alter a byte or copy a block of bytes from one area of RAM to another. It is good value at £3.50 plus 50 pence for postage and packing.

Bug-Byte ZXDB disassembler can be used in conjunction with its ZXAS assembler and occupies 4K from address 16514. It works entirely in hex and does not dump to the printer, although you can circumvent that by disassembling 12 lines or so, then calling 0869H, which is the Sinclair ROM COPY subroutine.

Another disadvantage is that some of the mnemonics belong to the 8080 rather than the Z-80, e.g., LD A, (HL) appears as LD A,M. It has a very large number of sophisticated monitor functions, so it is included in the monitors. ZXDB costs £6.50.

MicroGen Debug is also a disassembler with some monitor functions, works entirely in hex, and can be used with a printer. Care has to be taken when transferring from Basic to Debug and back, or the ZX-81 will crash. The monitor display is impressive, although more detailed instructions would have been helpful. The program lives above RAMTOP, which it resets automatically, and costs £3.95.

ACS Debug can be used in conjunction with the ACS assembler and disassembler to provide a complete, if rather expensive, machine code writing package. It stores above RAMTOP, uses decimal numbers only, and does not dump to the printer. Once again, it costs £5.50.

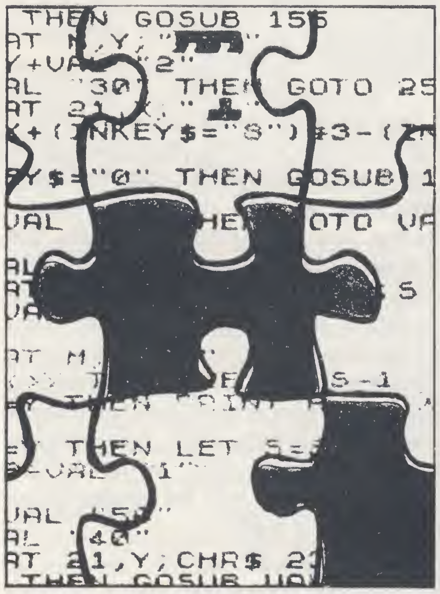
The Picturesque ZX-MC is another sophisticated monitor; it is rather like a separate operating system. You cannot use it with any existing machine code programs, as it uses low memory and has its own stack in high memory. It has its own loading and saving routines which operate at twice the speed of the ZX-81. The program is complete with a comprehensive manual for £7.50 but its incompatibility with other programs seems to be a serious disadvantage.

The Taurus Machine Code Monitor is placed above RAMTOP and is available as a cassette and also in EPROM form as part of its 16K RAM pack system. With the latter you can switch from 14K RAM plus 2K monitor to the full 16K RAM. Apart from the usual functions, it has a helpful hex calculator and a facility to create REM statements of any length. It also has a comprehensive manual at £7.50 in cassette form, and the RAM-pack system costs £48.

ACS has also produced an assembler and dissassembler for the Spectrum. The assembler is Ultraviolet. It contains the usual functions of an assembler and pseudo-instructions including a routine to insert ASCII characters in a specified region of memory. The program also allows the use of labels to refer to memory locations to which a user may wish a machine code program to branch. The program is very easy to use and should cause no difficulty for anyone who has just started to learn about machine code programming on the Spectrum.

Infrared is a dissassembler, also from ACS. The program is just a dissassembler and includes no routines which could help a user debug machine code programs.

Picturesque was one of the first companies to offer a Spectrum machine code monitor. It includes dissassembly and debugging routines and can be used on a 16K and 48K Spectrum. The program will dissassemble any area of memory, including the Basic ROM. The ZX printer can be used with it to print-out dissassembled code and lists of hexidecimal numbers. The monitor can also be used to display the CPU registers after a breakpoint has been made in the machine code routine. That will show the state of the registers after certain operations have been performed on them and is useful when debugging a program. When Picturesque releases an assembler which can be used with the monitor, the user will have a very powerful machine code writing tool.



ACS, 7, Lidgett Crescent, Roundhay, Leeds LS8 1HN.

Aylesbury ZX Computer Club, 12, Long Plough, Aston Clinton, Aylesbury, Bucks.

Bug-Byte, 98-100, The Albany, Old Hall Street, Liverpool L3 9EP.

Campbell Systems, 15, Rous Road, Buckhurst Hill, Essex IG9 6BL.

dK'tronics, 23, Sussex Road, Gorleston, Great Yarmouth, Norfolk.

Hewson Consultants, 7, Grahame Close, Blewbury, Didcot, Oxon OX11 9QE.

JRS Software, 19, Wayside Avenue, Worthing, Sussex BN13 3JU.

MicroGen, 24, Agar Crescent, Bracknell, Berkshire.

Picturesque, 6, Corkscrew Hill, West Wickham, Kent BR4 9BB.

Taurus, 47, High Street, Baldock, Herts SG7 6BG.

Education uses
Low-cost teaching

Sinclair Research has succeeded in its campaign for inclusion in the Micros in Schools Scheme. The Government funding scheme is intended to encourage the introduction of computer power to schools at a fast but sensible rate. Restricted previously to secondary schools and involving only the BBC micro and the Research Machines 380-Z, the scheme is to be extended to primary level and to include the Spectrum.

The MIS scheme has always had its critics, among whom are, of course, Sinclair users. It has been claimed frequently by teachers that, despite the exclusion of Sinclair from the original scheme, there are more ZX-81s in use in schools than all other micros combined.

Cost is the reason as much as anything else; one can buy six ZXs for the price of the cheapest version of the machines in the scheme.

The inclusion of the Spectrum in the Government list of machines approved for support in primary schools is, of course, a victory for Sinclair and a recognition of commonsense arguments.

If that implies that Sinclair micros are not suited to use at secondary level and above, that implication can be refuted by looking at the educational uses of micros at all levels. The question to air must simply be whether the ZX range can cope with educational needs. If the machines can, their remarkable cheapness is an added bonus, in that several can be used instead of one larger machine.

Educational uses can be broken.-down into a number of categories:

Computer awareness — the aim being to maximise familiarity with computers, their use, their uses and their abuses. Many schools include at least a few hours of such work in general courses followed by all pupils.

Computer studies — the use of computer hardware as the apparatus of the formal examination-orientated computer science teaching.

Computer-assisted learning — in which computer power is one of the many resources available for maximally-effective teaching in most subjects.

Administration — with the computer aiding the general running of a school, as in the case of any other business.

Data capture and process control — again helping, as in any other relevant business, with equipment interfacing.

Computer awareness concerns the provision of courses and experience designed to make all future citizens familiar with computers and their strengths and weaknesses. It is surely potentially the most important aspect of work in primary schools, as well as at secondary level; all other applications affect only the minority of pupils but this is essential for everyone.

It is also important in the sense that it can, and must, involve a large proportion of the staff of a school; at the moment there are surprisingly few teachers who know much about computers.

Computer awareness almost explicitly excludes hands-on experience and programming skills. Many schools with no computer power available have developed successful schemes of this nature in the last few years. It is a pity that happens and now there is no need to teach about computers without showing one in use.

Which one to show in use? The 1K ZX-81 may be programmed to emulate every aspect of computing one may wish to discuss in awareness classes. The 16K ZX-81 or Spectrum, with printer, can do so even better and in more depth but the programs need more effort to write.

Courses of that level need to deal with such things as the principles of data processing, the databank, number-crunching, interfacing. Computing in situations such as hotel booking, banking, office work and the laboratory can all be modelled adequately and cheaply by the ZX micros.

Perhaps the best way of consolidating computer awareness is for the majority of teachers to use computers as a resource in their teaching.

That takes us to the vitally important area of computer-assisted learning. The success and significance of Sinclair micros may be gauged from the facts:

There are about 25 software houses specialising in ZX learning products; two of them, AVC Software and Rose Cassettes, are well-established and successful.

The educational MUSE software Library — administered by EZUG — contains some 50 ZX cassettes — after only six months more than are available for any other micro.

The first software and the first publication in the Spectrum market were both for education, both from AVC Software.

The majority of those products can be classified as computer-aided learning material; between them they offer assistance in 12 school subjects. EZUG has estimated that a complete library of CAL programs covering the needs of children aged from five to 16 would need to contain upwards of 5,000 titles.

Any assessment of the validity of a given micro for education must rest in particular on its ability to cope with the formal needs of formal computing courses at CSE level and above.

Those needs are two-fold. First, the students have to undertake a fairly lengthy programming project, assessed as up to one-quarter of the marks for the examination. Second, they must have equipment available with which they can practise and explore the various concepts studied.

There is little difficulty with the project; the student must demonstrate his programming ability and that will be restricted, to some extent, whatever machine he uses. This year, for the first time, ZX-81s have been used for computing projects even at advanced level.

The equipment involved needs to be easy to use and readily-available and must include a printer. The ZX is by far the best in that context as it meets the requirements so cheaply that schools can readily have a fair number of keyboards available. The aim must be at least one between each two students.

If one spends money on ZXs for formal school projects, one must be sure that they also meet the needs of the theoretical components of the course.

The ZX-81 is only just about suitable there. Its major lack is the ability to handle files meaningfully but its omission of direct READ ... DATA, arguably unnecessary in real life but part of school syllabuses, and easy entry at machine-code level have also been quoted as precluding its effective use.

The ZX-81 can emulate those facilities adequately enough to allow the necessary demonstration and exploration. Certainly many schools had no choice but to use this micro as the basis of much, if not all, of their practical work.

Much closer now is the dawning of Computer-Assisted Freedom in education. Thanks to Sinclair, CAF may soon be a reality.

Dave Sayers looks at educational software

Helping brighten school days

THE NUMBER OF microcomputers in schools is growing rapidly in both the secondary and primary sectors. The increase in the use of micros in education has been stimulated by two things — the part funding of the purchase of computers by the Department of Education and Science and the many keen and interested heads of school and teachers who have introduced computing into their classrooms, often using their own machines and programs.

Yet there lies the irony; while machines were being bought there was little or no software available commercially which was written specifically for school use. The programs reviewed show that the need for software has been recognised and is being filled.

There are several points which have to be borne in mind; loading the programs has to be made as simple as possible and instructions should be clear and concise, in booklet form where necessary, with an explanation of what the program does and how it is operated, since instructions included as text are often sparse and uninformative.

The programs should run without bugs, use graphics if possible, and have interesting content together with excellent error-trapping.

Rose Cassettes sent six tapes for review, Junior Maths 1 and 2, Junior English 1 and 2, Champion Quiz and Arithmetic for the under-eights. Junior English 1 and 2, Champion Quiz and Arithmetic for the under-eights. Junior English 1 and Champion Quiz are all quiz-type programs and, as such, because of the limited content and vocabulary, largely unsuitable for regular classroom use. Champion Quiz, however, is better than most of its type. Junior Maths 1 includes long multiplication and division, fractions 1 and 2 and two other programs. The level set is difficult but children are taken through each stage of working the answer.

Junior Maths 2 includes areas, perimeters, sets and Venn diagrams, as well as two other programs. The standard of the tape is really very high, with good diagrams and instructions. It was, moreover, a winner with children.

Arithmetic for the under-eights is a superb cassette, dealing with addition, subtraction, multiplication and division. The level can be set at units; tens and units; or hundreds, tens and units. The numbers are shown in large characters, well-suited to a classroom monitor, and carrying is illustrated on-screen, in stages. A really good, worthwhile tape for classroom use, well put together and, like all the Rose software, properly error-trapped.

Junior Maths 1 and 2 and Arithmetic for the under-eights look like essential buys for any school using the ZX-81 in the classroom. It is hoped they will also be available for the Spectrum, too. The tapes cost £4.50 each and each cassette runs itself from loading.

The Fun to Learn cassettes, available from W H Smith and direct from Sinclair by post at £6.95 per cassette, were the next to be examined. Eight titles were reviewed, English literature 1 and 2, History 1, Mathematics 1, Music 1, Inventions 1, Spelling 1 and Geography 1.

Spelling 1 is an interesting idea which unfortunately is not suitable for school use. Words are played from tape, listened to by the child and the child's spelling of the word is checked by the computer. The only trouble is that there are too many sets of words on each side of the cassette, 15 to be precise, and the time involved in searching for sets, as well as the fixed nature of the words, renders it unsuitable for schools. Nevertheless, any parent interested in helping a child with spelling should consider it.

Geography 1 is very good indeed, with two programs — cities and countries in Europe and towns in England and Wales. When run, the relevant maps appear, although

borders in Europe are not marked. An atlas would help. The children enjoyed it and it seemed useful to them, as it can teach them as well as quiz them.

Mathematics 1 is a fairly ordinary mathematics tester. The Rose cassettes do the same job better, we feel, although this one deals with decimals as well.

Apart from Spelling 1, Geography 1 and Maths 1, the other programs are simple quizzes. That again renders them, on account of content and age level, unsuitable for schools. Geography 1 is very good value for money and highly-recommended. All the programs were well error-trapped, although they did not run themselves from loading.

ZX-81 Software, published by Brian Negus of 19 Westfield Drive, Loughbrough, Leics, LE11 3QJ, has three music teaching cassettes on offer.

The cassettes were well error-trapped, two of them dealing with the various clefs and the third with beats and note times. The four clef programs cover alto and tenor/bass and treble. When run, notes appear one at a time on a short bar, which is suitably large for use on a classroom monitor; the graphics are very good. The children are then quizzed on the notes shown. A help facility is included.

Beats aims to teach children rhythm by having them follow moving graphics. In mystery beats the graphics are static and the children have to discover the beat for themselves.

As in Arithmetic for the under-eights, the large graphics fit these programs well for use in front of a class. At £12 for three they must rank as good value, essential for music teachers. They also run themselves from loading, a good feature. Negus informs us that the programs will soon be available for the Spectrum.

AVC Software of PO Box 415, Harborne, Birmingham, B17 9TT submitted two programs for review, Tables Countdown and Geography Hangperson, at £3 per program. The programs are claimed to be completely error-trapped but one girl still managed to crash one when she typed RUN NEWLINE instead of GOTO 50. That was because, in using RUN, she cleared the variables loaded from the tape. AVC should reconsider that part of its programming policy, as one has to re-load the program to re-start it.

Tables Countdown is slightly expensive for what it does, as the Rose cassette arithmetic for the under-eights offers four programs of better presentation for £4.50, although a simple graphics reward is included. It tests the four rules, without help, at four levels of difficulty.

Geography Hangperson tests knowledge of locations in the British Isles with the aid of graphics clues. The locations are not well-thought-out and the contents of the program are limited but the clues are amazing — "Is there a town that makes frisbees, Sir? It looks like a man with his head on the wrong way

round".

We cannot recommend either of those two programs for school use. There are better programs available and at lower cost. Although the AVC programs can produce a print-out of scores, so can the Rose software, of which the children thought very highly.

Solent Software Engineering Ltd, of Brookvale, Waterworks Road, Otterbourne, Winchester S02 2DP produces the Zeta range of educational software, which Solent states has been fully-tested and developed by middle school teachers in Hampshire.

Eight titles were provided for review, on high-quality cassettes which ensured first-time loads; 16K RAM is required.

Each program is literature-supported with comprehensive operating instructions, a description of the game, how it works, what it does, and how to re-start it after a break. Full marks for such a good documentation. Graphs draws bar charts of the class birthdays. It ran well and, like all the other Zeta programs, is very robust. We would, however, have liked to see a more general graph-drawing program, with provision for copying the results to a printer.

Depth Charge and Zilog are ordered triple (X,Y,Z) and ordered pair (X,Y) co-ordinate games. In Depth Charge, a submarine has to be hunted while in Zilog an incoming space-ship must be destroyed.

Spell Invaders tests words set by a teacher, up to a limit of 10. A word is displayed, then blanked-out; the child has to "spell" it by shooting letters of the alphabet, Space Invaders style. Three incorrect letters only are allowed.

It is a very good game, which would have been even better with some means of printing-out a certificate with name, score and words tested on it.

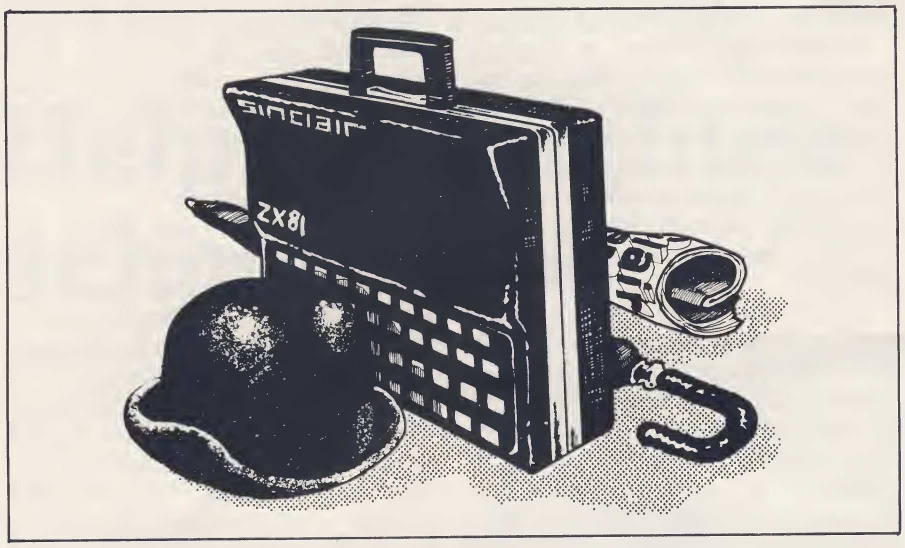
Oxo-, Oxox and Oxo + are simple mathematics games based on noughts and crosses, with sums displayed on the board, two children playing against each other to win the game by solving the problems and thus getting a row of their symbols.

Reaction Test is a computer familiarisation game in which letters are displayed and the appropriate keyboard letter has to be pressed. The response of the children is timed.

At £5.75p each, or £42 the set, they may seem a little expensive. That is not so. Solent Software states in its literature that "where the original purchaser is a school or college, additional copies of the programs may be made for use on other computers only within the said school or college."

Considering that they worked well, and taught specific things, the set of programs would be a good addition to the software library of any school, particularly as you would not have to reply on only the one set of tapes;

Loading in all cases was first-class and proves that good cassettes are vital for easy loading. Cheap tapes will not suffice. Apart from the tapes from Solent, instructions were poor for all the programs we reviewed. Educational software should be supported by comprehensive instructions in booklet form.



Mike Salem on business uses

ZX-81 could be in good company

BOTH SINCLAIR computers, the ZX-81 and the Spectrum, are classified as home computers. They are good for playing games and learning the rudiments of programming but have little to offer anyone who wishes to make more serious use of a machine.

While not having the power of other micros, like the Apple or the Commodore Pet, there is a growing number of people who think the ZX-81, and now the Spectrum, have much to offer people who wish to use the machines for a practical purpose, like owners of small businesses.

With the Spectrum only just beginning to be delivered in any numbers, the business uses to which it has been put so far are limited. That is likely to change as more become available, when it will begin to build on the foundations laid by the ZX-81.

It is cost which attracts most people to the ZX-81. The minimum requirement for serious work is the ZX-81 plus at least 16K of RAM. The printer is extremely useful. A television set and a tape recorder must, of course, be used. Such a configuration would have cost tens of thousands of pounds a few years ago. The ZX-81 and RAM cost about £80. Adding the printer and 48K of memory increases the cost to slightly less than £180.

The main limitations, compared to other micros, are that the ZX-81 does not support data files, that the printer is rather basic and that the keyboard is inconvenient to use. In addition, the machine does not allow data to be stored on and read from tape. The whole program, with data, must be loaded and saved.

Even if it were possible to read directly from a tape cassette it is such an inconvenient medium that it would be impracticable. The ZX-81 is limited to what it can hold in memory at one time. Whether that is a serious problem depends on the particular application.

Consider, for example, the payroll of a small company. A comprehensive program can be written to fit comfortably within 16K, with nothing omitted; in fact, a feature not to be found in payrolls on much more costly systems can be incorporated — gross pay and deductions can be worked out from the nett pay.

Room is left for full data on 30 employees to be held in RAM; there is no need for data files of any kind.

While that would not be suitable for a large company, it can save a small company doing payroll manually a great deal of unnecessary work. Another application is a what-if? type of program; a sheet of inter-dependent data is set up and then the consequences of changes are explored. For example, the effects on company profits of different assumptions about inflation can be assessed.

Despite the width of the printed output and the type of silver paper used, the quality is surprisingly good. It reproduces very well on an office copier. All the information which can be printed on a wider printer can, if necessary, be reformatted to use the 32-character width.

The ZX printer is not, of course, as good as a printer which costs more than the complete 48K ZX-81 system but it is a matter of inconvenience rather than impossibility. The graphics capability of the ZX printer is good.

The keyboard is perfectly usable, although slow and inconvenient. Keyboards of conventional type can be bought from independent suppliers for reasonable prices.

It is very easy to plug disc drives and various interfaces into a computer such as an Apple II. The ZX-81 is much more difficult to expand and interface. Nonetheless, many people are spending a great deal of effort on the problems and unbelievable products are appearing at unbelievably low prices.

The ZX-81 has a very good version of the Basic language. Numeric information is handled well; text information is handled better than by most machines, including some costing more than £10,000. The machine is small, light and very robust; 16K of memory goes a long way, as will be seen when discussing possible applications; 48K is unbelievable.

There have been problems with the interconnection between the ZX-81 and the RAM pack — the machine would suddenly go dead, losing all the contents of memory; those problems can be solved in an entirely satisfactory way. It is perfectly feasible to operate a ZX-81 for 24 hours a day.

The prime advantage is, however, the price.

A ZX-81 with 48K of RAM can do virtually everything a 48K Apple with printer but without discs can do — but compare the prices.

The price difference extends into the software field — compare the price and performance of a really good ZX-81 program to that of a similar program for a more

expensive machine. Any computer which is to be used in real applications must be reliable. The ZX-81 has a less than perfect reputation in that respect but all the problems can be solved.

Essential requirements for any computer are that all calculations must be done correctly; that the contents of memory must be stored without being corrupted or vanishing entirely; that it must be possible to store programs and data reliably and to load programs and data generated by a properly-set-up system.

The correctness of calculations is almost entirely dependent on the program. It is possible for hardware design errors to occur which cause odd problems but they are usually found and corrected by the manufacturer relatively soon after the release of a new machine.

The ZX-81 had such a problem; early machines did things such as 3 — 0.00000000000001 = 7 and 0.25**2 = 3.1423844. That was soon corrected and the old machines were repaired. Even the giants of the computer industry are not free from such problems.

Computers also have idiosyncracies which must be understood by programmers; many computers can obtain a number such as 3.9999999 by ading 0.1 + 0.1 + . . . repeatedly. If the program then tests whether the answer is 4, or takes the integer part, the result will not be what is wanted — that is the programmer's responsibility.

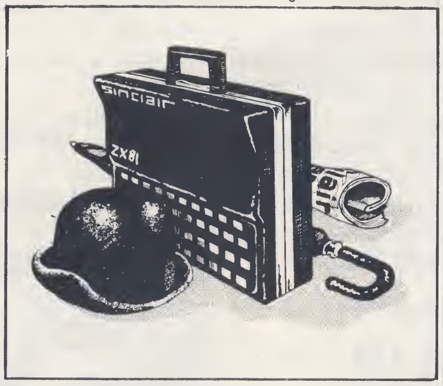
It is easy to design a computer without the particular feature just described but that adds to the cost or decreases the capacity to store numbers. The ZX-81 has a very good version of Basic. The handling of text information, in particular, is better than most micros, including some very expensive ones.

The integrity of data depends on the design of the machine and the reliability of components and connections. The electrical design of the ZX-81 is, in general, good; it is vulnerable, however, as are most microcomputers, to strong electrical interference on the mains supply.

The components used are generally satisfactory. The occasional rogue memory chip appears — in Sinclair and other computers. Usually the problem is intermittent. The chip is perfect when tested but causes trouble on warming-up. The presence of a faulty chip may be revealed by, say, incorrect letters appearing here and there in a program, or by the appearance of program lines which are not valid Basic — invalid lines cannot be keyed into the ZX-81.

In such a case, a faulty RAM pack usually must be exchanged or repaired. Rogue chips usually make themselves known fairly quickly. It is also possible for tapes to be produced which load perfectly with some recorders but which cause program corruption similar to that produced by faulty memory chips — or simply refuse to load — with others.

That is more common with duplicated tapes than with recordings direct from the ZX-81. In the short term, reliability is improved by loading at the highest possible level. In the longer term, such tapes must either be re-recorded by the user or



replaced by the supplier. The choice is the user's.

The most serious problem with the ZX-81 has been the sudden loss of all the contents of memory. That is due almost invariably to the connection between the ZX-81 and the Sinclair RAM pack. The RAM pack plugs into the ZX-81 but has two feet; when the keyboard is used the RAM pack and the ZX-81 move slightly, relative to each other. One of the 43 connections may be interrupted momentarily, with catastrophic consequences for the contents of memory.

The best solution is to use a RAM pack which employs a tight connector of high quality with gold-plated contacts and which does not in any way touch the surface supporting the ZX-81.

The ZX-81 can be made to operate very reliably. The expense and trouble involved in not great — certainly far less than that involved in living with an hopelessly unreliable system.

Company software

Taking a firm line

THE VARIETY of business software on the market for both ZX computers seems to be almost endless, as well as being confusing.

It is difficult for a beginner to decide which program is best for certain applications without becoming too involved with the technicalities of the business world. Business programs should teach the user who is also a beginner, as well as being an aid for the hardened businessman.

Hilderbay, run by Mike Salem, produces programs to cover a diverse area of business applications which can be used in the home and at work. Salem stresses that the programs were made as simple as possible to use.

One program which would be useful to anybody interested in buying a house, taking a loan, or working-out VAT, is Financial Pack 1. The cassette has three programs on it — Loan, Mortgage and VAT.

Mortgage deals with bank and building society. It has two main uses. The first is seeing the effect of small changes in the mortgage rate on payments before deciding to take the mortgage. Second, it is possible to discover how much has to be paid in interest at the end of the month. The program will calculate either the term or the instalment of the mortgage, depending on what information is available at the time.

The loan program will compute any of the terms involved in loan repayment — principal, interest, number of instalments — depending on the other factors available.

With VAT you may enter prices with or without VAT. The program will display price, VAT totals, and VAT rate. Financial Pack 1 costs £5.

Another program from Hilderbay is Budget. The program will help the user to keep track of expenses incurred during the year. It is possible to keep expenses under 50 headings.

Information on each item of expenditure can be entered but only the monthly running totals are kept.

The program will produce a bar chart of expenditure, budget plan, or difference between the budget plan and expenditure to date. The amount spent can be compared to the budget plan for the month. The program with data can be saved on tape in the usual way. Budget costs £5.

Finally from Hilderbay there is Payroll. The program will keep a record of employees and their pay for a small company. Payments can be entered on a weekly or monthly basis. After entering all the employee data and altering anything which needs to be changed, the program will calculate the present payroll.

The program includes a security routine which allows only a person who knows the correct code name to enter the program.

Investment on the stock market is becoming more and more popular. Micromega has produced a program which will keep track of a user's portfolio of stock and share investments. It is called Comp-U-Share. It monitors the most important factors measuring investment performance.

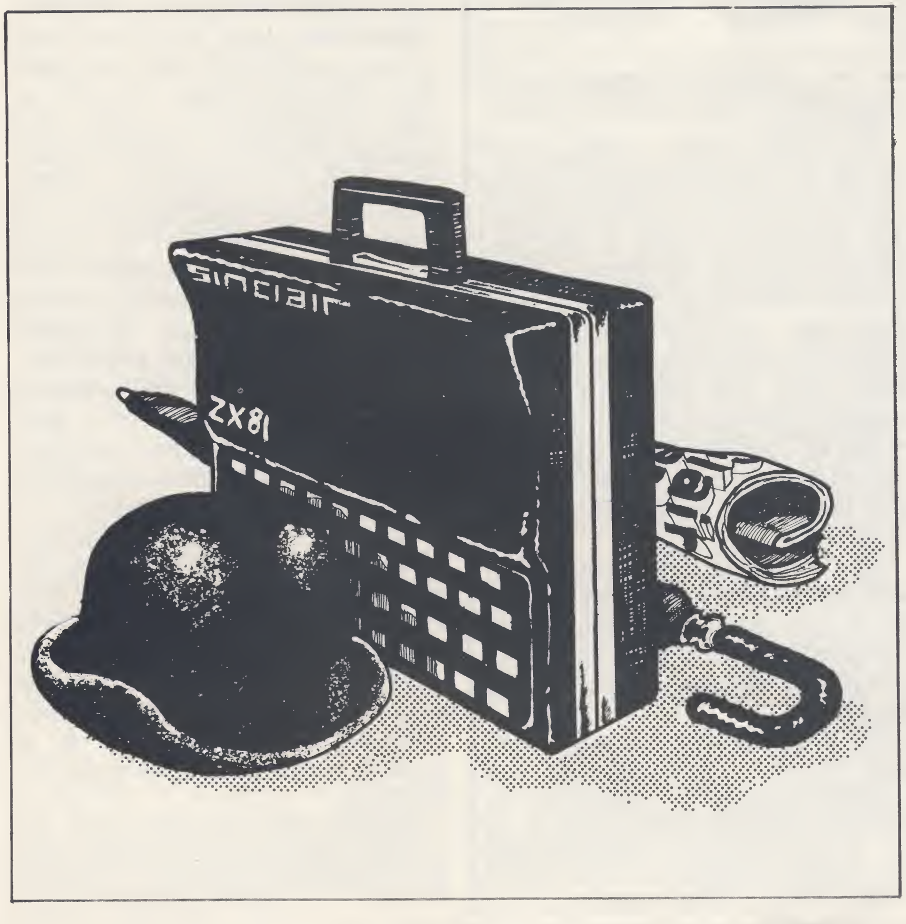
Once the relevant data is entered, it is possible to obtain reports on percentage gain and loss, price-earnings ratio, percentage nett gain or loss, and totals.

Records of shares can be indexed with numbers. When initiating a report it is possible to obtain an analysis of only those records which you desire by entering the relevant index numbers.

The program stores data files separately so that the user does not have to waste time storing both program and variables every time the program is used. It is also possible to verify the data stored on the tape in the same way as with the Spectrum. The program costs £9.95.

Micromega also produces a program to help with income tax. It may not decrease the amount paid but it can make the business of filling-in a tax form simpler. The program will run on both the 1K and 16K versions of the ZX-81. The 1K version is loaded in stages.

The Income Tax package has a mock-up of a tax form and a user guide which takes you through the business of filling-in a form with your tax data step by step. The program has been checked by chartered accountants and can be used for the tax year ending April 5, 1982. Income Tax costs £9.95 and there will be a 15 percent discount for updates.



Hestacrest provides a program called Accounts which will be of use to anyone preparing accounts from incomplete records. The program is menu-driven and when a client's income and expenses have been entered satisfactorily a profit-and-loss account and balance sheet can be displayed.

Various headings such as sales, stock, cash and VAT are specified in the program, using expense codes. The heading for each expense code can be changed to suit the user's purposes. The headings supplied take the ordinary sole trader as a model. Accounts is available from Hestacrest for £17.50.

Vu-Calc and Vu-File are two programs which were commissioned by Sinclair Research and produced by Psion. They are easy to use, extremely flexible, and convenient for displaying expenditure at the press of a button. The programs are part of a group of software cassettes sold by W H Smith alongside the ZX-81.

Vu-Calc can be used to display such items as income, expenditure and tax in a table displayed on the TV screen. It will also perform calculations on the data displayed from formulae entered to the table.

Vu-File can be used to store information such as club membership records, populations around the world, and time-tables. On the other side of the cassette is an example program which has in-store information on every country in the world. Vu-File and Vu-Calc are available for £7.95.

All the programs mentioned are for use only on the ZX-81, except those from Hilderbay which are also available for the Spectrum.

Almost every business application has been covered during the last year and the quality of it is very high — surprisingly high for such a small machine as the ZX-81. It has been proved that the small system many people thought was of no use for serious applications is far more versatile than they imagined.

Hilderbay, 8-10 Parkway, Regents Park, London NW1 7AA.

Micromega, 230-236 Lavender Hill, London SW11 1LE.

Hestacrest, PO Box 19, Leighton Buzzard, Beds LU7 0DG.

Sinclair Research, Stanhope Road, Camberley, Surrey GU15 3PS.

NOW AVAILABLE FROM J.P. GIBBONS A.I.B.

(Author of the best Banking programs in the world!)

The PBS Bank Reconciliation is believed to be the first such program to be available for the Sinclair ZX-81 and ZX Spectrum.

Just look at some of its features:

* Automatically matches bank statement items with transactions recorded in your 'Personal Banking System'.
* Lists all outstanding items such as uncleared cheques, funds received but not advised to you, Bank charges — highlights all discrepancies. No more 'scratching around' to try and find out why your statement balance is different to what you expected.
* Full search facility, allowing on-screen correction of any item.
* Saves and loads data to and from tape (ZX-81 version, at double speed). Exchanges data with the main PBS program. No need to save programs.
* Single key operation — uses machine code keyboard scan for single letter commands (ZX-81 version).
* Extremely easy to use. No jargon to wade through before you get started. Demonstration program included.
* All item lists, searches (with totals), can be output to your ZX printer.
* Full instructions included.
* Unique after sale maintenance provided.
* Operates in conjunction with the 'Personal Banking System' (full input and correction facilities plus automatic posting of standing orders on due dates and fully detailed statements).

So banish the headaches, make life easier for yourself, send £15 for the Personal Banking System + PBS Bank Reconciliation and Users Manual.

Existing clients need only send £5 for the PBS Bank Reconciliation which will operate with their present PBS.

The Personal Banking System (as reviewed in popular computing weekly — 1st July) is available separately for £9.95 for cassette and user manual. ZX-81 PBS owners can order the Spectrum version for just £5.

Please specify whether ZX-81(16K) or ZX Spectrum (48K).

Guaranteed software available from

J. P. GIBBONS A.I.B.,

14 Avalon Road, Orpington, Kent BR6 9AX

YOUR PBS IS NEVER OUT OF DATE!

THE ADVENTURE GAME THAT'S FOR REAL !!!!

£6,000 PRIZE! FREE HIT SINGLE! 48K ZX SPECTRUM

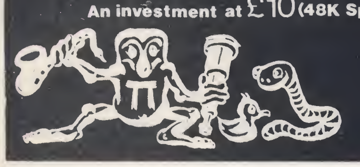
(16K ZX81 VERSION NOW AVAILABLE)

Will you be the first to locate the Golden Sundial of Pi in time and space, and be rewarded with the original? Exquisitely crafted by the winner of the De Beers Diamond International Award, from gold, diamond and the most precious of the earth's riches.

PIMANIA - where saxophones turn into hangliders, where music meets madness and where the Pi Man rules supreme! He'll talk with you, he'll befriend you, he'll betray you, he'll even do the Hokey Kokey! Animated cartoon graphics! Full musical score! Spectacular colour and sound effects! Includes free hit single "Pimania", with vocals by Clair Sinclive and the Pi-Men!

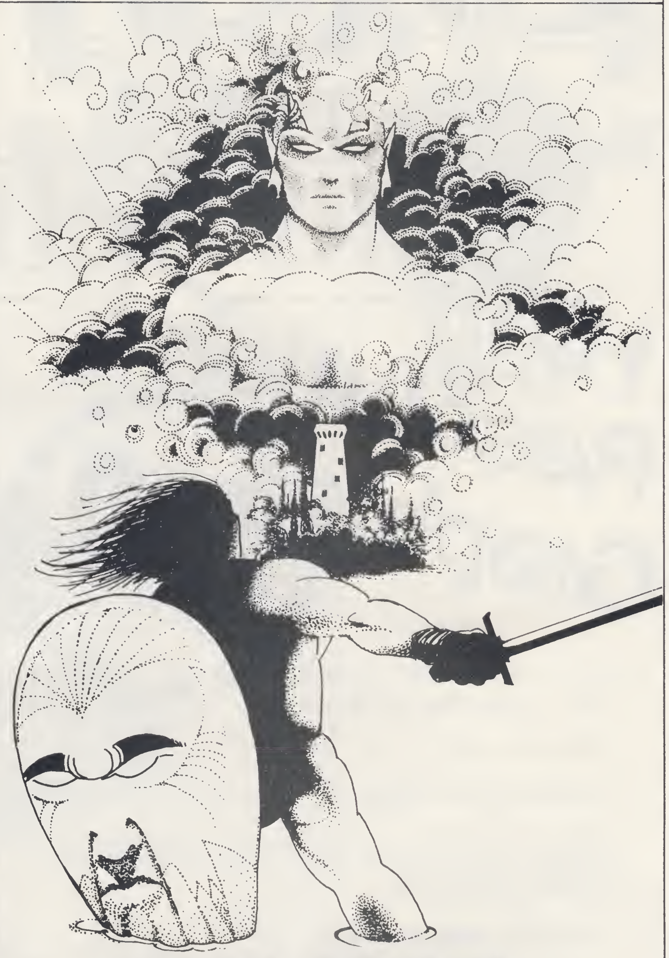
It could take you a week to play, it could take you a lifetime! PIMANIA, " the best evidence that computer gaming has come of age . . an adventure enthusiast's dream!" (Computer & Video Games)

An investment at £10 (48K Spectrum) £8 (16K ZX81)



Automata Ltd. (S)
65a Osborne Road
Portsmouth PO5 3LR
England

SPECTRUM 48K : ZX81 16K
SIX PART ADVENTURE



BLACK CRYSTAL
THE QUEST IS ABOUT TO BEGIN

The ultimate role-playing adventure for the SPECTRUM/ZX-81. You can become a warrior, elf or wizard on a quest to find and use the rings of creation: to destroy the Black Crystal and defeat the Lords of Chaos. Held within six programs lies a land of fabulous treasures and mythical monsters. Journey through the land of Beroth, explore the castle of shadows, descend into the Shaggoths lair, search for diamonds in the sea of sand but beware of sand sharks! Confront the fire demon in his temple, battle against the Lords of Chaos and win your way to the Black Crystal.

By splitting Black Crystal into six programs we can provide more variety and detail than any other adventure for the Sinclair computers.

Real time monster battles. Superb graphics. Save game feature. All six parts are supplied together on cassette, boxed with instruction booklet.

SPECTRUM 48K: 180K OF PROGRAM IN SIX PARTS — ONLY £7.50
ZX81 16K: OVER 100K OF PROGRAM IN SEVEN PARTS — ONLY £7.50
WHY PAY MORE FOR LESS OF AN ADVENTURE

CARNELL SOFTWARE

To: CARNELL SOFTWARE, DEPT 2, 4 STAUNTON RD, SLOUGH. SL2 1NT.
Please send me: Black Crystal for my,

Spectrum 48K................ £7.50
ZX8 16K..................... £7.50

I enclose a cheque/postal order (payable to Carnell Software) for £ ________

NAME...

ADDRESS......................................

....

PROPRIETORS: R. CARNELL, S. GALLOWAY

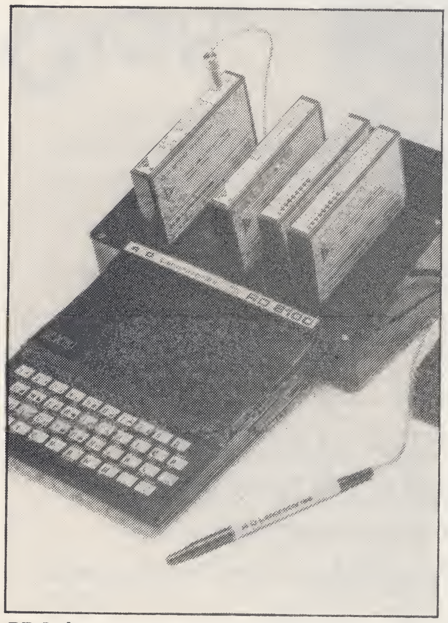
HARDWARE WORLD

Many people have found the ZX-81 and Spectrum too limited for their purposes and looked for ways in which their machines could be expanded. That demand has been met quickly and now there is a large range of goods which can be obtained at reasonable prices and which will do everything from expand the memory to improve the graphics. Stephen Adams, our hardware reviewer, selects some of the better items available.

Economical ways of improvement

Stephen Adams reviews the add-ons

The Kempston keyboard



RD Labs expansion unit

THE FIRST THING people groan about with a basic 1K ZX-81 is the lack of memory. Even on a 16K Spectrum you have only 9K of memory in which to program. So the first hardware add-on most people buy is more memory. The largest memory is not available from Sinclair but from two companies, Memotech and Downsway.

The Downsway is smaller, cheaper than the Memotech, but the Memotech can be used with other memory-mapped devices by switching-out the 8K-16K memory in two 4K sections. Only 56K is available anyway — 48K for Basic and 16K for machine code, because of the way Sinclair designed the ZX-81.

The Sinclair 16K RAM pack is no longer economical to buy and the cheapest is from Econotech. It is only a bare board with no case but does the job well at £19.95.

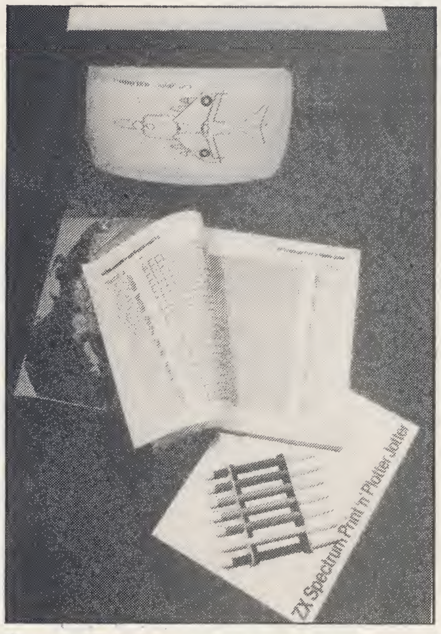
If you have suffered the dreaded RAM pack wobble which crashes your programs you will be delighted to know of at least two devices which will stop it, even though most RAM packs have better connectors than the Sinclair ones. A metal plate sticks under the ZX-81 RAM pack and has a hook at the back to prevent the RAM pack from moving about. It is available from Microware. The second is a flexable ribbon cable which fits between the ZX-81 and the RAM pack. It is produced by d'K and Kade at £10.

For the 16K Spectrum, with sockets, there is the 80K RAM board by East London Robotics which gives two 32K banks of memory switched by an OUT instruction and 32K boards by d'K. The Sinclair board is not economical as the Spectrum has to be sent back for fitting.

Another way of extending your memory is to use floppy discs to give fast, endless memory. The Macronics system works as a data array to hold programs or data for £160 with the ZX-81.

EPROMs are useful for storing programs but the only EPROM programmers of which we know for the ZX-81 — none for the Spectrum — are from Orme and Lander and the both work in hex.

Next on the list seems to be a replacement keyboard or case for the ZX-81 user; even the Spectrum



Print 'n' Plotter

ZX99 tape control

user now has a choice. Keys which move are vital to give the user a response when he presses a key and a BEEP or repeat key facility are also very useful.

Top of the chart must be the Kempston keyboard which replaces that of the ZX-81 and is only ¼in. higher. If you prefer a real typists' keyboard, then the Computer Keyboards one with stepped keys and 30-degree angle must be your choice. The only repeat keys we would recommend would be those which fit under the ZX-81 keyboard as they work with any keyboard, Kempston and Haven Hardware.

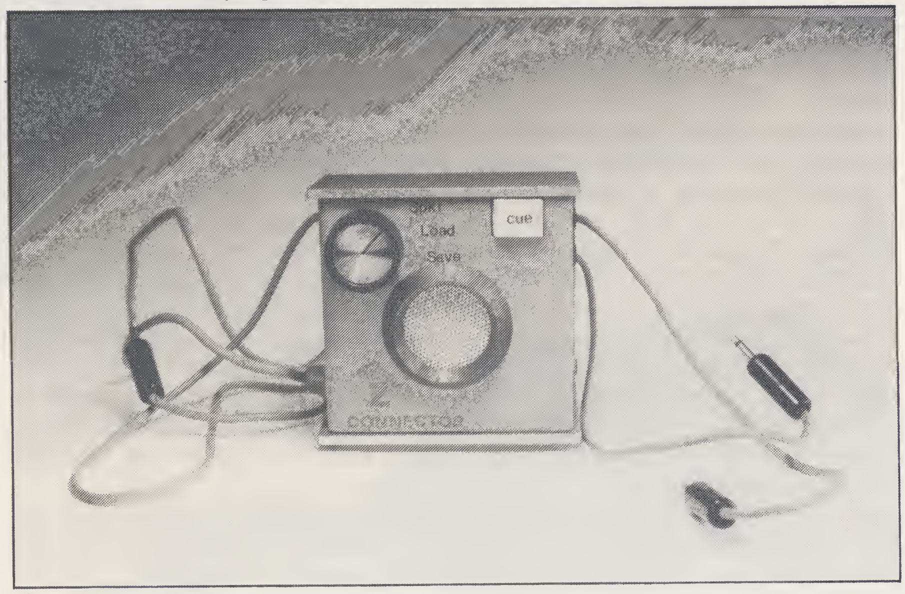
One case to mention, though, is the Fuller, which provides a light keyboard and case containing power supply, ZX-81, RAM and extra boards. For a system in a case, excluding a TV set, Custom cases are available through Sinclair agents at Camberley, Surrey for the ZX-81 and the Spectrum.

A large, non-portable system based on the ZX-81 for the business user has been put together by Cobra, using standard metal slides to hold the various boards — up to 12 in a case with keyboard, discs and hi-res graphics. RS232 printer and other sockets are available.

Ports seem to be a favourite way of getting more from your computer, as they are the only connection it has with the outside world. DCP is probably the most popular for the ZX-81 user as its "P" pack not only provides an input and output port which is memory-mapped but also 4K of extra memory. To those ports can attach voltage-to-number converters (A/D) or D/As — other way round — for joysticks, computer-controlled robot arms and many other things. Memory-mapped ports are easier to use as they can be controlled from PEEK and POKE in Basic.

On the Spectrum, however, I/O-mapped ports must be used. On the ZX-81 that would require the use of machine code but on the Spectrum that has been provided as IN, OUT commands in Basic. Kempston and William Stuart are good examples of I/O mapped ports Stuart has it as part of a sound board. Bolton and Technomatic are good examples of a cheap, bare board memory-mapped port and for a large system — up to 14 boards — the RD8100 by RD Labs would seem to be best choice. Most ports require a motherboard or some form of extra connector for, like the RAM pack, this shuts-off all expansion for Microdrives. That must be included in the cost if you are thinking of buying a port.

Memotech 64K memory expansion



The OZ tape connector

The chunky graphics Sinclair provides with the ZX-81 are satisfactory for making pictures and other designs, if you plan them beforehand. The Print-n-Plotter jotter makes full use of this, as pictures can be copied from magazines or the design for a new game layout can be done on paper. With the Spectrum and its fine plot — 176 × 256 — it is essential, rather than wasting time writing a good program which looks bad.

Character generators are available from d'K and others which allow you to program an alternative character set from the Sinclair one; most, however, need some connection inside the machine. Lower-case letters and proper Space Invaders are then possible on the ZX-81. On the Spectrum it is all provided, which caused a slump in the add-on market for the Spectrum.

Mention must be made for the hi-res board from QS, as that makes the graphics on the ZX-81 the same as the Spectrum, except that it is in black and white. The board is expensive but it fits on to the back of the machine and every dot on the screen can be plotted individually.

Inverse video can provide a pleasing change from the glare of a white screen on a ZX-81. The best and simplest is from D Fritsch, as it also sharpens the display. Another way of reducing the glare is by applying a coloured screen from Ellenbee across the TV screen.

Sound on the Spectrum is

somewhat limiting, as BEEP stops the program temporarily. The sound of the BEEP is also rather weak but can be amplifed by a sound box from Fuller or by putting it over the TV loudspeaker via a Compusound adapter. A sound box is available for the ZX-81 and the Spectrum which can provide three tones and a noise facility. It is called the Zon-81 and is made by Bi-Pak. It contains its own loudspeaker and volume control.

Keyboards also need a BEEP if they are to indicate to the user that he has pressed a key. The best seems to be the KAT from TV Services which fits inside the case and works with any keyboard. It gives two bleeps and is also very useful for checking tapes, as it gives a BEEP at the start and finish of a program. A BEEP is also available from Fulcrum which is solderless and fits in the same place as the KAT.

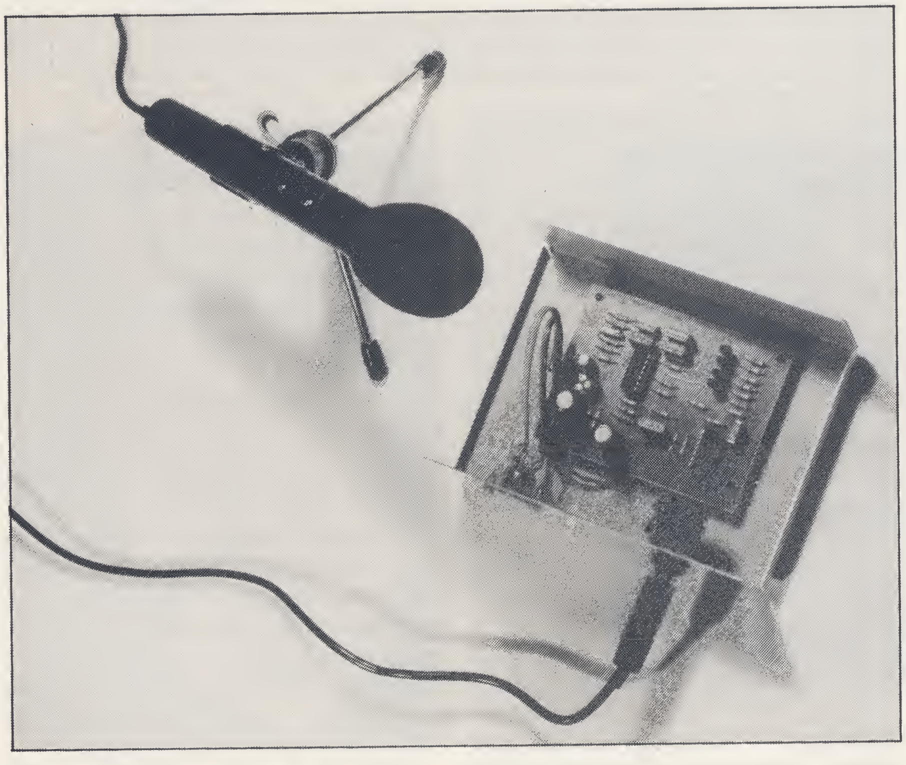
Speech input is something unusual for a personal computer, especially for one as cheap as the ZX-81. The Big Ears system by William Stuart must therefore be mentioned. In its present state it is a little too sensitive but several people have had good results with it. It records one second of speech and then tries to match it to voice patterns stored in its memory. It is the only one of its kind at £49.95 plus VAT, so why not try it?

Speech output is a little easier and two devices can are available for the ZX-81 user, one from DCP and the other from Namal. Both have ROMs which can speak numbers and measurements in an American voice through a built-in loudspeaker. The Namal one, however, can be programmed to produce any type of voice or word.

Although machine code programs are available from firms like Picturesque, a tape control device which controls the tape recorder, too, had not been available. The ZX-99 by Data-Assette can control up to four tape recorders, has an RS232 printer interface and can save data as well as programs. If you want to reduce the feedback by unplugging one of the cassette leads, why not try Abacus or a manual switch of Stephen Adams for the RZ1, a program-controlled one?

Q-Save by PSS is a program hardware package which allows you to save and load tapes at 4,000 baud — 16K in 29 seconds with an ordinary tape recorder for £15.95 on the ZX-81.

If you have problems with your tapes, Mike Salem at Hilderbay may be the man to help. He has produced a book on getting the best from tape recorders, including tips on unloadable tapes. Also he markets a test tape and meter for checking tapes.



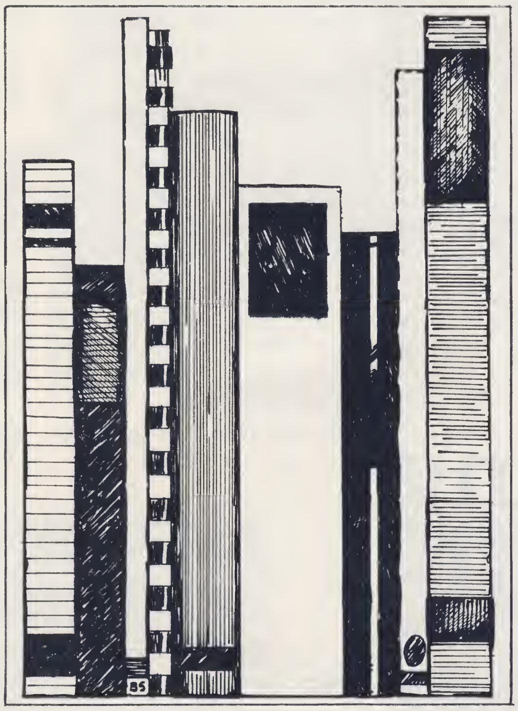
Big Ears speech input system

DCP Speech Pack

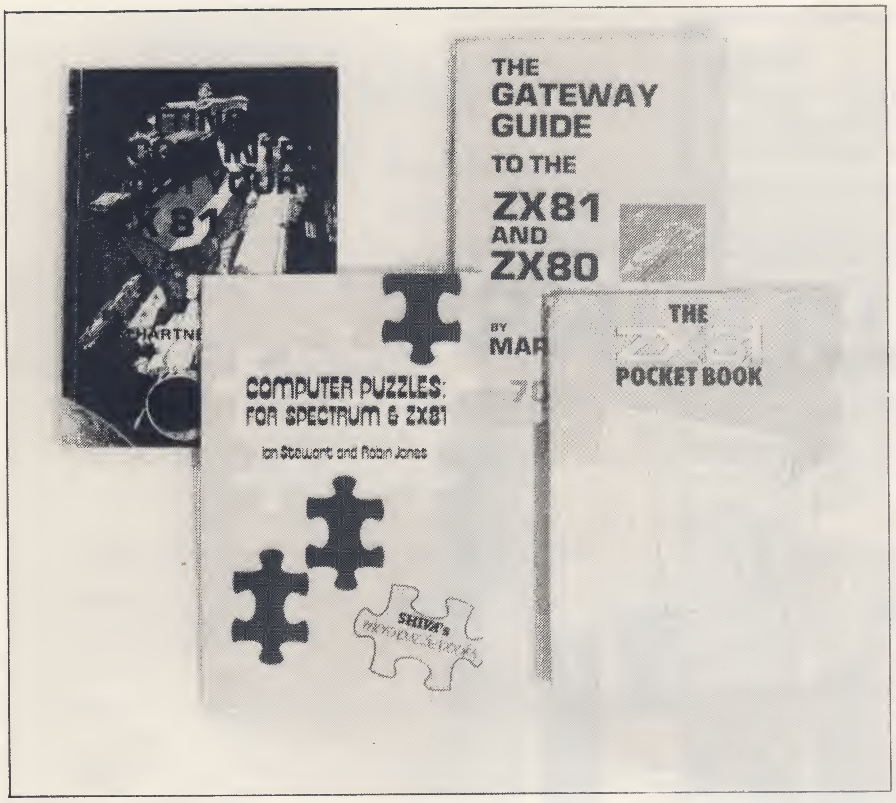
Suppliers

Stephen Adams, 1 Leswin Road, London N16.
Haven Hardware, 4 Asby Road, Workington, Cumbria C14 4RR.
Abacus Electronics, 186 St. Helens Avenue, W.Glam SA1 4NE.
Bi-Pak, PO Box 10, 63A High Street, Ware, Herts.
Bolton Electronics, 44 Newland Drive, Bolton B15 1DP.
Cobra, 378 Caledonian Road, London N1 1DY.
Compusound, 32 Langley close, Redditch, Worcs.
Computer Keyboards, Glendale Park, Fernback Road, Ascot, Berkshire.
Custom, JBS, Stanhope Road, Camberley Surrey.
Data-Assette, 44 Shorton street, London NW1 6UG.
DCP, 2 Station Road, Lingwood, Norwich, Norfolk N13 4AX.
d'Ktronics, 23 Sussex Road, Gorleston, Gt. Yarmouth, Norfolk.
Downsway Electronics, Downsway House, Epsom Road, Ashtead, Surrey.
Econotech, 30 Brockenhurst Way, London SW16 4UD.
Ellenbee, 11 Lichfield Close, Chester-le-Street, Co Durham.
East London Robotics, 14, Darwell Close, East Ham, London.
D Fritsch, 6 Stanton Road, Thelwall, Cheshire WA4 2SH.
Fulcrum, Hillside, Steep Lane, Findon, Worthing BN14 0UF.
Fuller, ZX Centre, Sweeting Street, Liverpool 2.
Hilderbay, 8-10 Parkway, Regents Park, London NW1 7AA.
Kempston Electronics, 60 Adamson Court, Hillgrounds, Beds.
Lander, 32 Clockhouse Road, Romford, Essex RM5 3QJ.
Macronics, 26 Spiers Close, Knowle, West Midlands B93 9ES.
Microware, 131 Melton Road, Leicester.
Memotech, 3 Collins Street, Oxford OX4 1XL.
Mamal Electronics, 25 Gwydir Street, Cambridge CB1 2LG.
Orme, 2 Barripper Road, Camborne, Cornwall.
Print-n-Plotter, 19 Borough High Street, London SE1 9SE.
PSS, 112 Oliver Street, Coventry CV6 5FE.
Quicksilva, 92 Northam Road, Southampton SO2 0PB.
RD Labs, 5 Kennedy Road, Dane End, Ware, Herts SG12 0LU.
William Stuart Systems, Dower House, Herongate, Essex.
Technomatic, 17 Burnley Road, London NW10 1ED.
Taurus, 47 High Street, Baldock, Herts SG7 6BG.
TV Services of Cambridge, Chesterton Mill, Cambridge.

BOOKS



Sinclair computers are ideal as cheap machines on which to learn the capabilities of computers and the basics of programming.
To help in the learning process a small publishing industry has grown up, producing a wide range of books for the novice and the enthusiast.
John Gilbert takes some off the shelf.



Expanding bookshelf

John Gilbert looks back at the year's books to help enhance the ZX machines

IN THE LAST year scores of books have been written about Sinclair machines and it seems as if anyone who knows anything about the ZX-81 or Spectrum can enter the literary field and become an author.

During the early days of the Sinclair boom, most authors were content to collect together a few programs and write a short piece about each one. That made a book which could be on the streets in the shortest possible time.

One of the main publishers of ZX books is Interface. Its early contributions to the ZX-81 market included *Getting Acquainted With Your ZX-81*, by Tim Hartnell, and *The Gateway Guide to The ZX-81 and ZX-80*, by Mark Charlton. Both books are full of listings and very little explanatory text and similar in content.

Hartnell seems to have made a one-man attack on the book market in the last year. His first books were full of listings and contained little text but his latest have dealt more with the techniques of programming and are more helpful to a computer user.

The rationale for books of listings was that they would give people who had just received their machines something to type-in and play with before beginning to program. That reason no longer applies. Magazines such as *Sinclair Programs* print listings which first-time users can type-in, so there is no longer a need to fill the pages of a book with long listings.

The lesson does not seem to have been learned. Several companies have published books of program listings for the Spectrum. They are well-produced and colourful but ideal only for first-time users or people not interested in programming. The best of them include *Computer Puzzles for Spectrum and ZX-81*, by Ian Stewart and Robin Jones from Shiva, and *Over the Spectrum* from Melbourne House. *Over the Spectrum* is a book of listings but the introductory text to each program is more comprehensive than in other books.

Details are also given about how to adapt techniques used in the programs to the user's programs. The Shiva puzzle book is exceptional. The authors pose problems and give listings which simulate the problems in an effort to solve them.

At some time in the last year authors must have felt that the market for books of listings had reached saturation point. People wanted to learn how to program and several books on programming techniques appeared.

'One of the first on techniques was *The ZX-81 Pocket Book*, by Trevor Toms, published by Phipps Associates. Toms described several ways of saving memory, a much-discussed subject among ZX users. In the book are several programs which use the techniques described and also provide light entertainment when difficulties arise.

Another book on techniques which caught the imagination of users is the *Sinclair ZX-81 — Programming for Real Applications*, by Randle Hurley, published by Macmillan. It details several software projects, including a word processor and data filing program. Hurley also deals with memory-saving techniques and problems which may be encountered by a user when programming. The author has promised a book on machine code but it has not been published.

Peek, Poke, Byte and RAM, by Ian Stewart and Robin Jones, and published by Shiva, gained fame for providing an easy introduction to programming for the beginner. The book takes nothing for granted about the reader and contains many things which the ZX-81 manual did not embrace.

The author of a technical book faces two problems. The wording must be precise but not fall into jargon. The author must also try to find new areas to explore and not draw on material used in other books. That does not seem to worry many authors but soon the technical

market could be saturated with books on the same structured programming techniques and memory-saving devices.

People need to know about those things but there are already many books which explain them adequately. It would be better if authors were to explore other areas and there are many as yet unexplored ones in computing.

The machine code market for Sinclair machines is still very young but what has been published is very good. One of the easiest books about machine code is *Machine Language Programming Made Simple for your Sinclair*, published by Melbourne House. The book takes an almost child-like stance but proved very readable and excellent value for the beginner.

Mastering Machine Code on your ZX-81, by Toni Baker, is also a good introduction to the subject but slightly more difficult for the beginner to understand.

Machine code can be entered into the Spectrum with greater ease than the ZX-81 and is expected to make people more willing to use it. Once they start doing so they will have greater programming power at their fingertips. The Spectrum is an ideal machine on which to learn and if this sector of the book market is overlooked it would be a pity.

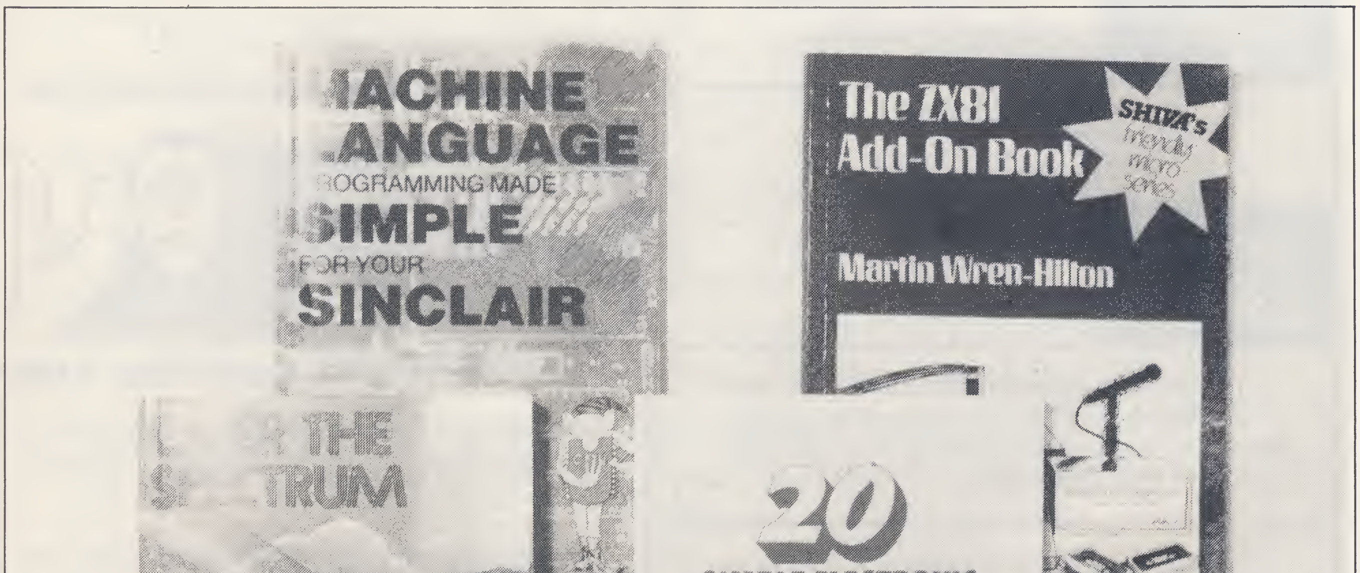
A new market for ZX books has been opened in the electronics field. Two examples are the *ZX-81 Add-on Book*, by Martin Wren-Hilton, published by Shiva, and *20 Simple Electronic Projects for the ZX-81 and other computers*, by Stephen Adams. It shows how to build things like tape recorder controllers, numeric keypads and even a cheap thermometer. The book explains the working of several add-ons at present available from retailers and includes projects which the reader can set up, including a burglar alarm and a voice recognition system.

The book market is becoming as lively as the software market and no doubt many books will be produced for the Spectrum and the ZX-81 in the months to come.

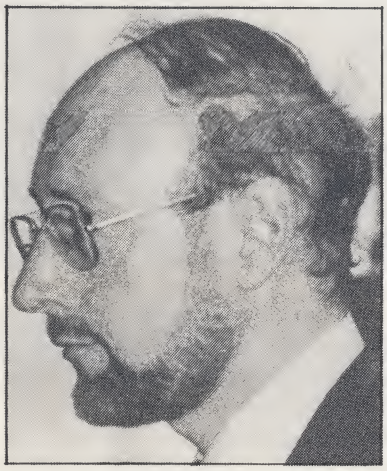
A general criticism of all the books mentioned is that they are all too costly. Paperbacks which would sell for no more than £3 are being offered at £9.

There are two probable reasons for the high prices of many books. Perhaps the market is so small that the prices have to be high to cover the costs. The other reason could be that the publishers know that people will buy a recommended book at almost any price. The satisfactory solution seems to lie between the two.

The book market needs strict controls over price and quality if it is to survive in the computer field. If companies do not start to reduce prices and publish better-quality books, computer users will become more cautious and selective. That has already begun to happen with software and the signs are that it is starting to happen with books.



INSIDE SINCLAIR



During the year we made a number of behind-the-scenes visits to Sinclair Research, speaking to people involved closely with various important developments in the company.

Much of what they had to say has proved interesting in the light of later developments. For example, Nigel Searle, head of the computer division, said it did not occur to him in June that Boots, Currys or Rumbelows would sell a computer. Boots and Rumbelows now sell the ZX-81.

In July, Richard Altwasser and Steven Vickers two of the leading figures in the design of the Spectrum, were keeping their plans secret after leaving Sinclair Research. They have since launched a major competitor in the home computer market, the Jupiter Ace.

We begin in May with a visit to the Timex factory at Dundee to see the manufacture of the ZX-81.

Lynd Church at Timex in Dundee

Production goes like clockwork

IN KEEPING with its philosophy of sub-contracting all manufacturing processes, Sinclair Research appointed Timex to produce the ZX-81 when it was introduced in March, 1981.

Since that time, production at the Timex plant at Dundee increased from 10,000 to 60,000 units per month, while production of the Sinclair printer, also in the Timex factory, is 15,000 units per month.

David Chatten, production controller with Sinclair, explained that although the company does not undertake its own manufacturing, it is very thorough in its analysis of prospective component manufacturers. "We get people who are good at manufacturing particular components, then get everything assembled in a good production plant", he said.

The rate of growth of units produced at Dundee not only reflects the success of the ZX-81 but also justifies Sinclair confidence in Timex, a company which had little previous experience in the assembly of electronics equipment.

David Chatten

Allan Johnston

Allan Johnston, project manager of board assembly at Timex, explained that the company decided to expand into new technology when watch technology began to change. He said: "Our experience at Dundee was in mechanical watches but development in quartz and digital technology led us to consider expansion into other areas".

So Timex and Sinclair began discussions on the production of the Sinclair flat-screen tube and Microvision pocket TV, for which Sinclair has announced a £5 million four-year capital investment programme. Because Timex had no experience in electronic assembly, it was decided that the production of the ZX-81 would be a first step in the learning process, added to which Timex was prepared to make the capital outlay for equipment needed for such assembly.

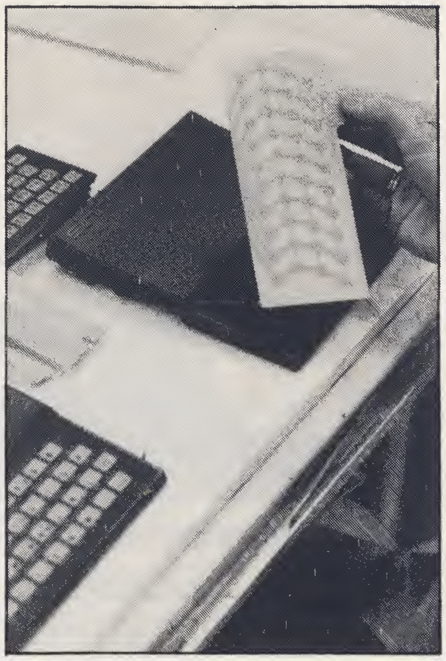
Chatten also stressed that assembly staff at Timex are of very high calibre, the best of whom were, initially, taken off-line to train in ZX-81 assembly. Chatten added:

"Getting Timex to do the assembly on the ZX-81 may have been a risky decision in the short term but in the long term it provides us with considerable security, giving us the opportunity to build good working relationships before full production on the pocket TV begins".

Whatever the risks, production of the ZX-81 seems to be proceeding smoothly, with only minor difficulties still to be solved, according to Chatten. The process starts with the build-up of the printed circuit board, including the addition of diodes and resistors. The board then goes through a flow-solder machine, which solders all the components into place and includes a cutter to tidy the leads.

It is only at that point that the chips are added to the board. They arrive at the Timex plant in anti-static tubes and are fed into a slide dispenser to ease the process of manual assembly. It is at that point that the first test of the equipment is made.

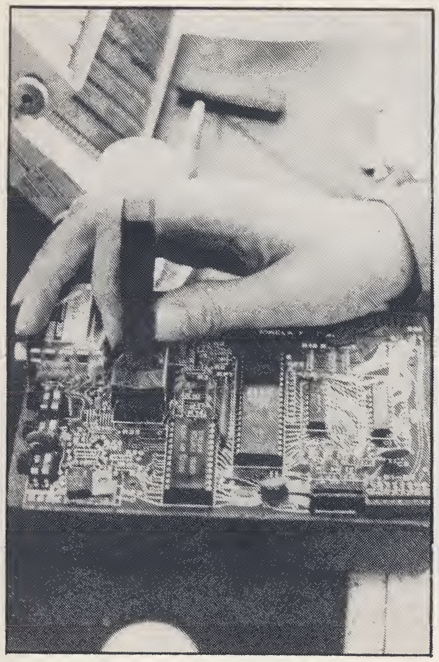
Building-up the circuit board



Assembling the keyboard

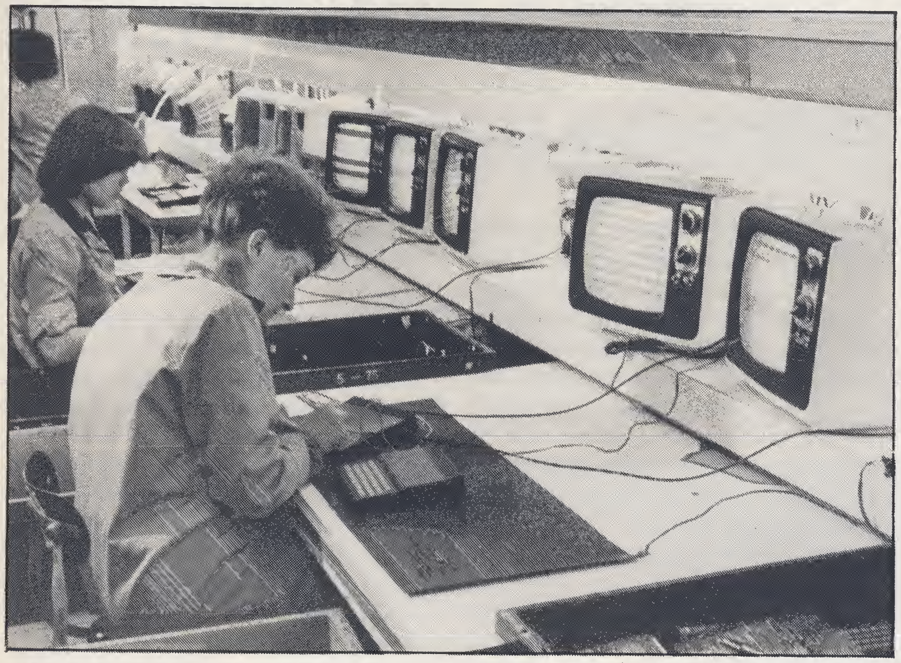
Bearing in mind that the assembly process has not then reached the keyboard stage, the test is to establish that the board is alive and well. It is called the K test, because all it requires is the appearance of the K on the screen, notifying the user that the machine is ready for instructions.

If a board fails the K test, it is then tested on a GenRad 2270 which identifies the fault and produces a



Inserting the chips

Testing for faults



Testing the keyboard

Finished printers

printout denoting the area of fault. The printout is then attached to the board, which is returned to the rework section, before going through the entire process again.

Assuming that the board has passed the K test, the top half of the casing with the touch-sensitive keyboard and connectors is assembled. At that point, the ZX-81 is again tested, running off a test program on cassette which will display a series of OKs on the screen for every key tested.

Finally, the bottom half of the casing is attached and the completed ZX-81s are packaged for distribution.

Timex has agreed with Sinclair to market personal computers, peripherals and software in North America.

Distribution for them will be handled by the established Timex shipping agents, distribution network and retail outlets in the U.S., though Sinclair will continue to sell the ZX-81 by mail order from its Boston-based subsidiary until Timex sales reach agreed targets. At that stage Sinclair will begin to concentrate on sales of its yet-to-be-released flat-screen TV.

The machines in North America will be marketed under both Sinclair and Timex names and Sinclair will receive a royalty on all sales under the agreement, which covers current and future Sinclair personal computer products and Timex developments of Sinclair computer technology.

Timex obviously has a fair amount of confidence in the product it is assembling for Sinclair, demonstrated by its use of the ZX-81 to test the printed circuit board of the Nimslo 3D camera, assembled by Timex and soon to be marketed in the U.S.

Sinclair believes that such confidence is well justified, pointing to what it considers to be a low percentage of returned equipment. The

return figure on the ZX-81 is 2.4 percent, though the percentage return on kits is higher, at around 13 percent.

Chatten claimed that the higher figure of kit returns is due more to poor component insertion and bad soldering than to component failure.

He admitted that of the 2.4 percent returns of supplied ZX-81s, around one percent of failure is due to power supply faults, an area in which Chatten believes there could well be improvement. In fact, it is Clive Sinclair's stated aim for a return percentage of one percent overall and it is partly to that end that he appointed a quality assurance manager, David Fuller.

While an aim for continued improvement in machine reliability is a necessary part of successful business practice, the gamble of engaging a company with little experience in electronic assembly seems to be paying-off. Chatten commented:

"There is some room for improvement in the assembly process — perhaps greater use of automated production lines — but on the whole we are very pleased with Timex work".

On the production line.

Linda Bishop talks to Nigel Searle

Plotting course for company growth

THE launch of the Spectrum is only a part of the development plans of the Sinclair Research computer division. The company also intends to market a full range of peripherals and software for all its computers and to expand overseas and into the educational market.

The man behind the ambitious expansion plans is the head of the division, Nigel Searle.

"I expect the Spectrum will not replace the ZX-81 but will sell alongside the ZX-81 and be the beginning of a range of computers", he said.

"They will be fully-supported in terms of peripherals and software. We have already developed a mass storage device which is of our own design and that will be announced later".

He added that in future Sinclair intends to launch new computers with a full range of software.

Of the educational market, Searle said: "Many schools have a ZX-81 but the price of them is such that many schools ought to have 20 or 30 of them. We hope to penetrate that market in the U.K. and elsewhere".

That ties with plans for overseas growth. The company is in the middle of searching for foreign distributors.

"We expect our overseas sales to increase substantially", he said.

Searle has a long association with Clive Sinclair. He first joined him in the Sinclair Radionics company 10 years ago designing pocket calculators. He then moved to the U.S., first in California and later in New York, where he was responsible for promoting the company's calculators and watches.

He stayed with the company until 1977, when he left. "The calculator business was not doing too well and also it was not really the same company once the National Enterprise Board was involved", he said.

Two years later Clive Sinclair formed Sinclair Research, launched the ZX-80 and Searle rejoined him. He ran the U.S. office in Boston, concentrating on selling the ZX-80 and 81 until taking his new job.

Software is an area in which Searle is particularly interested. The company has begun a software development project which should build the library to 200 programs.

"They will be available only through W H Smith, 26 to start, which is just the tip of the iceberg, covering games, education and some business. It is an area we have neglected in the past but we have spent time getting together a wide

range of software for the ZX-81".

Searle was involved closely with the launch of the Spectrum and decided to continue the unusual Sinclair marketing strategy of concentrating on mail order.

"With minor variations we are launching our new products the same way we always have done", he said, but added that "there are no plans at present for putting the new machine into W H Smith, which is Sinclair's only retailer".

He said the reason was that "not many others are selling so many computers as we are. We have sold far more computers by mail order than anyone who has sold through stores".

He added that the original idea behind the mail order decision was that when Sinclair first went into the computer market there was no obvious retail outlet for a personal computer.

"It does not occur to me, or anybody else, that Boots, Currys, Rumbelows, would sell a computer.

"It also makes good sense financially to sell through mail order. We do not have to give a discount to retailers which you normally have to do".

The promotion of a new product through retail distribution can cost so much that the price of the product has to be raised by 50 percent.

Heavy advertising is still essential and Searle again adopts an unusual policy by not having a pre-determined budget.

"We are willing to spend as much on advertising as will produce a profitable number of sales", he explained. Last year the cost was slightly more than £5 million and in 1982 it looks as if it will be more than £10 million.

Where that is spent depends on the product, with advertising in the technical computer journals, particularly the magazines, and the Sunday magazines.

"So far we have had products which have been of interest to both the specialist computer market and the general enthusiast market but we might well have products in the future which would be of interest only to the serious computer user".

For the Spectrum, Searle is concerned not only with selling the machine but also with persuading people that is it better than rival products. "We would not introduce a computer unless it was significantly different from our existing one," he said.

His return to Britain has made life much more hectic than it was when he was in Boston selling the two ZX computers.

"So far working here has been a bit like jumping on a train which is passing at about 60mph. It seems as though there are many things to be done".

Jim Westwood

Backing Sinclair

ONE OF Jim Westwood's first pieces of engineering wizardry was the contraption which enabled him to carry-out soldering work from the comfort of his bed. Were it not for the fact that he was only 12 years old at the time, that might be mistaken for the sign of an extremely lazy character. As it is, it merely emphasises the trait of ingenuity which has helped him during his 20-year working relationship with Clive Sinclair.

During those two decades, he has had a hand in such innovative products as the Sinclair pocket calculator, the three more recent computers and the promised flat-tube TV, not to mention the transistor radios and hi-fi equipment of the early days.

Today, at 34, he is known as senior, or chief engineer, with Sinclair Research, a role which combines engineering and management skills.

It is a far cry from the early 1960s when he joined Clive Sinclair and one secretary straight from school and relied on trial and error, as much as natural aptitude, to take him through his first days as a technician.

"Engineering of a kind was always my hobby, even when I was very young. Wherever I went you could be sure of finding a trail of broken torches in my wake. I had to take everything to pieces and gradually I was able to put it together again", he says.

It is that consistent, if unorthodox, philosophy which has stood him in good stead for so many years and ensured that the products in which he had a hand were always at the forefront of technology.

"I think it must be unusual to find someone like me in a fairly senior position without formal training", he says modestly, "but when you are always working unconventionally, as we are at Sinclair Research, I don't think training matters very much. Aptitude is more important".

From his small office in Cambridge, surrounded by an orderly chaos of electronic equipment, he seldom works on fewer than three ideas at a time. Of those, few come to fruition and only a handful reach initial design stages.

"The most difficult part is deciding what we want to achieve in the first place. We start with a mess which we call a breadboard. That has a very basic outline of our concept.

"All of us here have electronics in our bones and so when we first discuss an idea we know roughly its chances. Because we always produce 'firsts' we can be reasonably sure there will be no competition.

"The real worry is always whether it will catch on. You might feel sure there is a certain demand in the market but you are never sure just how it will sell".

Westwood admits that he still flinches at the sound of some of Clive's ideas but adds: "It's a challenge managing to achieve something without using expensive components and I like that challenge.

"Of all the products with which I have been involved I think the ZX-80 is my favourite. It was a real breakthrough in the use of cheap components. It is something which ought to be in the Ark by now but I am still proud of it".

Westwood is a modest and unassuming man, dismissing his early role at Sinclair simply as a matter of "fiddling with the components and trying to get the thing working".

His confidence grows as he talks of Sinclair generally and it is clear that he recognises the combined talent in the company, a team which would be sadly incomplete without him.

"We are always surprised at how long it takes the rest of the world to catch up with us. After working with Clive for years, you learn that it is worth trying to do things other than the straighforward way. It has amazing benefits. All our products show imagination and inventiveness; they make other people envy us and want to work for us.

"We spent a long time getting all the people together and now we have a very strong team, which is one of the main reasons for our success, in my view".

Westwood, who is married to a former teacher and has four children under the age of 10, is adamant that his family will not be reared on a diet of TV games.

A seemingly bad advertisement, perhaps, for his work, but he is already introducing his children to the concept of computers as an aid to living — and they love it.

"My only adverse reaction to the whole thing is that the instruction manuals leave much to be desired when you are trying to teach children".

Aside from the sheer technology of his job, he has become involved increasingly in management, taking part in the decision-making and ensuring that ideas are carried through the system.

He enjoys decision-making and the follow-up process, including the field trials which, for the flat-tube TV, will take him round the world.

"There has not been a great deal fo travelling so far. Of course, I go to Dundee often and our private aircraft has made a huge difference to that; it beats the sleeper anyway.

"It will be another challenge to work on the field trials. We will have to set up small laboratories or take the equipment with us, trying it and perhaps modifying it slightly to suit the various surroundings".

Ask what follows the flat-tube TV and Westwood is overcome by a sudden vagueness, at odds with the forthcoming nature of the rest of the interview. He may be untrained, he may be shy, but Westwood knows when he is being tapped for a secret; and, like all good engineers, he is giving away nothing.

Steven Vickers and Richard Altwasser set up on their own

Searching for gold at the beginning of the Rainbow

TWO OF the leading figures in the development of the Spectrum have cut their links with Sinclair Research to set up their own company.

Richard Altwasser, who designed the hardware, and Steven Vickers, who wrote the programs for the ROM working memory, have formed Rainbow Computing Co. Apart from publishing a book of programs for the Spectrum, the company plans are a closely-guarded secret.

"It is necessary for us to be very cagey and apart from the one thing which we have announced, we would like to leave anything we are doing secret until it is ready for launching," says Altwasser.

They decided to make the move because their major project for nine months, the Spectrum, had ended and, like many other people, they wanted to be their own bosses.

"We had plenty of freedom working at Sinclair but at the end of the day the company was run by one man and if a decision needed to be made, there was one man who took that decision," Altwasser says.

He and Vickers add, jokingly, that they had also been tempted because of the money Clive Sinclair was making.

Altwasser, 25, gained a degree in engineering at Trinity College, Cambridge and went to work for a micro-based automation company at Worcester but found the organisation too limiting. After 18 months he left and joined Sinclair Research in September, 1980.

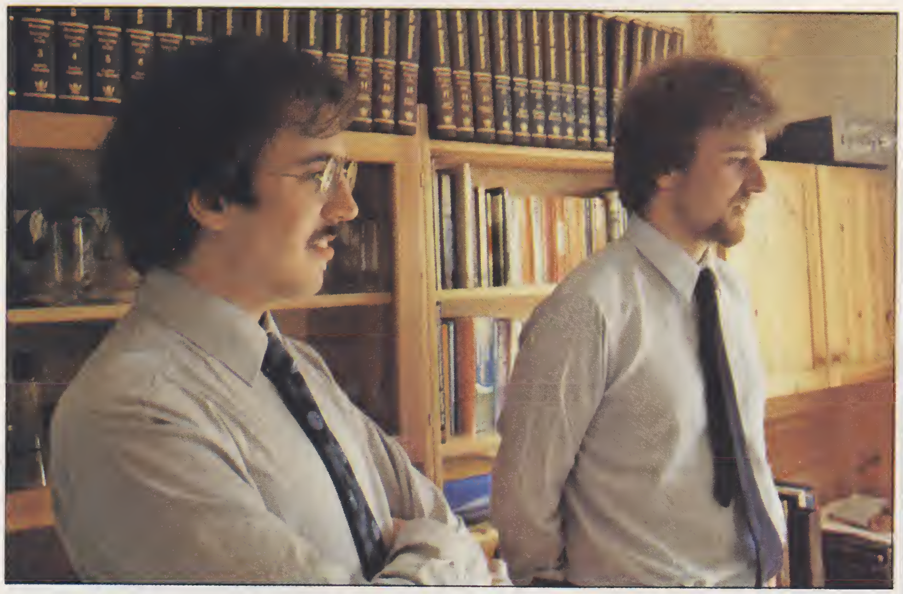
He did some work on the development of the ZX-81 and after its launch in 1981 he was made responsible for computer research, which involved him in the design of the hardware of the Spectrum.

Altwasser has also been writing software for the ZX-81 and his Cambridge Collection has sold 30,000 copies.

Before joining Sinclair he had a little knowledge of computing, owning a TRS-80 and having run a course in teaching Basic.

Vickers' knowledge, however, was much less. "Two years ago I did not even know what a ROM was," he says.

Vickers, 29, was also at Cambridge, gaining a degree in mathematics at King's College before doing his PhD at Leeds. In 1980, after writing for a job to a number of computer companies,



including Sinclair, he joined Nine Tiles, a software consultancy based near Cambridge, which had written the ROM working memory for the ZX-81.

His first job was the adaptation of the 4K ZX-80 ROM to make an 8K ROM for the ZX-81. He also wrote the manual for the ZX-81 and went on to write most of the ROM for the Spectrum, as well as assisting with the manual.

Both say that they found working for Sinclair very exciting — "providing you can cope with the pressure without having a heart attack." The main difference they found between Sinclair Research and other companies in electronics was that "deadlines were very real deadlines".

Development of the Spectrum was typical of the way in which Sinclair Research works. A rough specification was worked-out with the main requirements, including colour, high-resolution graphics and improved tape storage interface.

That was set in September, 1981 with a final deadline of the Earls Court Computer Show in April, 1982. By that time the Spectrum had to be ready to go into production, which meant that not only had all the development work to be done at Sinclair Research but also all the suppliers had to be chosen and the production lines at Timex had to be tooled-up.

That had to be done in conditions of great secrecy and very little information leaked about the machine, although Altwasser says he was surprised by how much was known about it before the launch. In the end, with many nights of working late, the deadline was met and the Spectrum launched on time.

Other benefits of working for Sinclair were that there was no shortage of money for research and, as it was a small company, it was easy to obtain quick decisions on new ideas and new ways of doing things.

"When I went for interview I asked about money being available if a piece of equipment was needed and was told that a request was never refused, but that they might advise about something which would be better," say Altwasser.

For the future, Vickers and Altwasser say they are concerned to prevent a Japanese invasion of the British market. Their plans for doing that are to remain secret.

Asked if their name denoted any link with the Spectrum, Altwasser replies that the only connection was that it has been one of the suggestions for the new machine which they had liked, so they had decided to use it.

One of their major concerns is that they should be able to keep pace with the latest developments in their field.

"There will always be the fear that something you have designed will be out-of-date as soon as you have finished it," he says.

They also think that the present generation of computer technologists will find increased pressure from today's schoolchildren. Altwasser says that teenagers are able to grasp ideas with which he had difficulty less than three years ago.

Rick Dickinson

Modesty blazes

AWARD-WINNING industrial designer Rick Dickinson is modest about his achievements, which so far include the ZX-81, for which he won a Design Council award, and the Spectrum.

"I don't think I have ever been delighted with anything I have done", says the blond, 26-year-old prodigy. "There always seems to be room for improvement".

Dickinson is a meticulous worker and while both the ZX-81 and the Spectrum are selling beyond all expectations, he adds: "I would never let anything go to production unless I was happy with it".

Graduating from the Newcastle Polytechnic pioneering industrial design course, Dickinson and his classmates are equipped, theoretically, to design anything "from knives and forks to ocean tankers".

Dickinson produced items as diverse as a chain saw and a road tanker during his first year as a qualified industrial designer, which he spent freelancing in Wales.

He had already spent some time working for Clive Sinclair while he was studying for his degree and it was not long before he was absorbed as a full-time member of staff and the company's sole industrial designer. He is responsible for the appearance of Sinclair products down to the layout of the components inside and the pattern of information on the keyboards. His membrane keyboard for the ZX-81 was revolutionary and largely responsible for the low retail price of the product.

Dickinson has learned that price is the ultimate justification and on all his designs he has to bear in mind the cost factor as well as the straightforward appearance of any item.

The membrane keyboard was a great success and Sinclair has had to cope with numerous pirate copies since its inception but, as with everything, it had its disadvantages. Its main disadvantage was its inability to register touch. To ensure

you have a response it is necessary to look at the screen — there is no reassuring click when you touch each key.

For the Spectrum, Dickinson has returned to a raised keyboard but again he produced a first by making it from rubber. He says:

"I like the Spectrum much more than the ZX-81. It was much quicker to design but much more complicated. It is a step upmarket and I was really trying hard for a super-smart machine. It is not for quite the same amateur market".

The process of design is a long one. Normally it begins when Clive Sinclair outlines his idea to Dickinson, including his demands about size. "He will resolve in his own mind the specifications and he will always say how small it has to be. I think how can it be that small? Yet he is always correct in the end and we produce something which seemed impossible to me in the beginning".

Armed with his brief, Dickinson then spends a few days with his sketchbook, exploring ideas, but he likes to begin work in three dimensions as quickly as possible and is soon modelling in Perspex or plasticine.

The next stage is to produce the finished model in Perspex but obviously it has no components inside — it is produced as a solid block.

That model is detailed, even down to the graphics which Dickinson has painted on. Layout of the interior follows, with the designer using all his powers of logic to ensure that each component is in the best possible place. Perhaps the most difficult part is the keyboard. Dickinson says:

"We spent a great deal of time on that. It is the only interface between the user and the product and it has to be right. We were trying also to cram on more information than anyone had ever done. I believe that form should follow function".

Design of the ZX-81 took about six months. The Spectrum was quicker but with all his major projects Dickinson also has to set aside time for add-ons to existing computers — the work is never finished. His main project now is the flat-tube TV.

His biggest problem with that is that Sinclair has already been working on it for some time. Normally he is briefed at the same time as the electronics engineers but this time the inside is already finished. It is also another first, which means Dickinson cannot research by looking at existing products in the field.

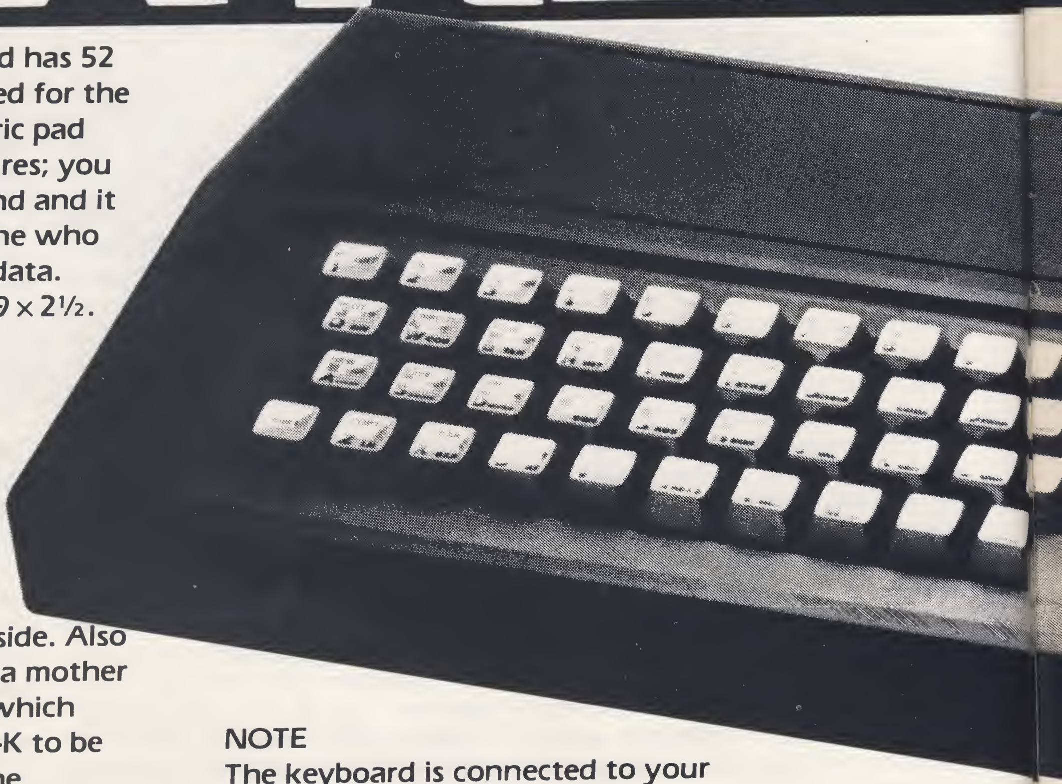
"That is the most exciting thing with this company, you know; many products are the first of their type, so you are really in on something new".

Dickinson is content with his life in every way. At school he liked the sciences and the arts and his job ensures that he remains involved in both. He spent one year on a foundation course at art college at Grimsby before starting to read for his degree and feels the experience was invaluable. He is happy with his work at Sinclair.

"We all work very closely, very much as a team. Most of the information is in people's heads. There is no time to be formal and put it on paper. It is a good atmosphere in which to work".

For this award-winning Yorkshireman, it abounds with opporttunity, too. He has entered the Spectrum for a Design Council award and on his drawing board are the initial stages of the flat-tube TV — another first, and possibly another award.

ZXKEY



Our new cased keyboard has 52 keys, 12 of these are used for the numeric pad. The numeric pad offers some useful features; you can cursor with one hand and it will be a boon for anyone who enters a lot of numeric data. The case measures 15 × 9 × 2½. The computer (ZX81 or spectrum) fits neatly inside. Also fitted inside the case is a mother board (81 model only) which allows 16K, 32K and 64K to be fitted in the case. All the connections are at the rear of the case i.e. Power Mic, Ear, T.V. and the expansion port. The power supply could also be fitted inside. This means you have a very smart self-contained unit.

NOTE
The keyboard is connected to your computer by a ribbon cable and this has connectors fitted which simply push into the Sinclair connectors. It is a simple two minute job and requires no electronic skills or any soldering.

ZX81 ARCADE SOFTWARE
THE FOLLOWING EXCITING MACHINE CODE ARCADE ACTION GAMES ARE AVAILABLE FOR THE 16K ZX81 ★ CENTIPEDE ★ METEOR STORM @ £4.95 each

- ★ 16K Ram Massive Add On Memory Fully assembled and tested £19.95
- ★ 64K Memory Expansion £49.95
- ★ 4K Tool kit full of utilities to aid the programmer in constructing and de-bugging

Fully cased with numeric pad	**£45**
Uncased with numeric pad	**£30**
Case	**£15**

dk'tronics

JOYSTICKS

ZX Spectrum / ZX81

22.95 INC.VAT & P+P

FOR ONE JOYSTICK AND INTERFACE MODULE

BUILT, TESTED & READY FOR USE

★ **NO SOLDERING,** plugs into rear expansion port between ZX and Ram Pack, Printer or Microdrives.

★ **TWO JOYSTICKS** connect via one interface module.

★ **NO SPECIAL PROGRAMMING,** Joystick 1 operates as keys 5 to 8 through inkey instruction.

★ **IMMEDIATELY COMPATIBLE WITH ALL CURRENT SOFTWARE** using arrow keys for movement.

★ **INTEGRAL 'FIRE' BUTTON**
FREE 'Video Graffiti' listing + full instructions.

To: A.G.F. HARDWARE, 26 Van Gogh Place, Bognor Regis, West Sussex PO22 9BY

QTY	ITEM	ITEM PRICE	TOTAL
	JOYSTICK	6.99	
	INTERFACE MODULE	15.96	
ZX Spectrum ☐ ZX81 ☐ Please tick		FINAL TOTAL	

ZX SPECTRUM SOFTWARE

SUPERDRAW16 £5.00

List of Features

* 16K Spectrum graphics pack.
* Full screen high resolution colour.
* Moving cursor control.
* Large alphabet facility.
* Pictures saved on cassette.
* Automatic "slide show" option.
* Menu driven, easy to operate, crash proofed.
* Documented to usual high Video Software standard.
* Demonstration slide show.
* Audio commentary on reverse of cassette.

SUPERVIEW48 £5.00

List of Features

* 48K Spectrum personal viewdata.
* Page creation with moving cursor.
* Large alphabet option.
* 48 full screen two colour pages OR
* 24 full screen full colour pages.
* Random page recall.
* Continuous page rotate.
* Print option.
* Full operating manual to our usual standard.
* Audio commentary on reverse of cassette.

**Prices include VAT, P & P.
Immediate delivery.**

Full range of ZX81 software still available.

**VIDEO SOFTWARE LTD.
Stone Lane, Kinver,
Stourbridge,
West Midlands,
DY7 6EQ.**

From Mikro-Gen for your Sinclair

Scramble (ZX81) £3.95
Scramble (Spectrum) £5.50
Frogs (ZX81) £3.95
Masterchess (48K Spectrum) £6.95
Bomber (ZX81) £3.95
Paintmaze (ZX81) £3.95
ZX Chess (ZX81) £6.50
Breakout (ZX81) £3.95
Sorcerer's Castle (ZX81) £3.95
Space Invaders (ZX81) £3.95
Asteroids (ZX81) £3.95
Tempest (ZX81) £3.95
1K Gamespack (ZX81) £3.95
Debug (ZX81) £3.95

All available on cassette, with library case, direct from the makers (please make cheques/PO's payable to Mikro-Gen and add 40p post & packing). Our range is constantly being up-dated – write for full details of the latest games and add-ons for your Sinclair.

MIKRO-GEN

TRADE ENQUIRIES WELCOME

24 Agar Crescent Bracknell Berks RG12 2BK
Tel: Bracknell (0344) 27317

SPECTRUM CLOSE-UP

The biggest micro news item of the year was the introduction by Sinclair Research of a colour and sound microcomputer, the Spectrum. While retaining many of the features of the earlier ZX-81, improvements were made in a number of areas. Two of our leading writers assessed its capabilities.

Colourful addition to Sinclair range

Robin Bradbeer impressed by value

THE SPECTRUM is a colour and sound computer for the incredibly low price of £125. At that price it undercuts the BBC Microcomputer Model A, its direct competitor, by around £175. In designing the new machine it is clear that the rejection of the Sinclair offer to build the BBC Micro was foremost in the company's mind. The specification is very similar and will certainly affect sales of the Acorn-based machine. It is as if Clive Sinclair has turned to the powers that be in the Government and BBC and said: 'I told you so.''.

The Spectrum is a small computer measuring 233 × 144 × 30mm, or slightly wider but not so deep as the ZX-81. The basic model has 16KB of RAM and 16KB of ROM. That compares to most other common computers for ROM but it is more RAM than most models in the less than £300 price range.

Another 32KB of memory is available at around £50 and that plugs into some sockets already built into the main board. The 48KB RAM model, therefore, is potentially as powerful as the very common Apple II computer costing around three times the price.

The Spectrum plugs into a normal UHF TV tuned to channel 36 and all characters are shown lower-case unless specified by using the capitals shift. There is a capitals lock, which is very useful. The Basic is based on that of the ZX-81 and some of the features lacking on the original model have been included in the latest one.

The screen can display 24 lines of 32 characters, although the individual pixels which make up each character — 64 in all — can be accessed and changed at will. That means that 256 × 192 pixels is available for graphics.

The screen format is very similar to the Commodore Vic, with a border area and the active screen within the border. At switch-on, the system automatically enters a mode where border and screen area, or paper as it is called, are white and the letters, or ink colour, black. That overcomes the strange effect noticeable on some colour computers where the border area is different from the working area, which makes the screen look smaller than it is.

The colours of the border, paper and ink can be changed easily with commands of the same name. Eight colours are available, although judicious use of the graphics characters available makes intermediate colours, like orange, possible. It is also possible to have 21 user-defined graphics characters, which will allow Greek or other alphabets to be used.

Unlike some computers built in the States, the Spectrum also has a £ sign on the keyboard. Everything can be printed on the ZX printer; the ZX-81 expansion memory pack cannot be used with the Spectrum.

Many commonly-used routines in the graphics are available automatically. For example, a circle can be drawn with the 'circle' command by specifying the centre and diameter.

In addition to the ink and paper commands, the Basic has brightness and flashing commands. Other useful graphics functions include an over command which allows characters to be super-imposed at any point. The six colour-control commands can be used over the whole active screen area or locally within each individual 8 × 8 pixel group which makes up each character.

Like the ZX-81, the plot command accesses one pixel at a time and the attributes of each block can be used to control the characteristics of that pixel. Colour control codes, which can be accessed directly from the keyboard, can be inserted into text or program listings and, when displayed, will over-ride the globally-set colours until another control code is encountered.

All control commands can also be used within strings and it is enter-

taining to define a string which has different-coloured characters and background colours in it. A simple print command using that string causes it to be printed to screen just as stored.

Editing is the same as for the ZX-81 but the addition of auto-repeat on every key makes editing easy, especially when the cursor is moved around a long line.

Some additions to the Basic include the means to enter a binary number directly. That is the method of generating the user-defined characters, of which there can be 21 directly-attributable to some of the keys on the keyboard. The 8 × 8 matrix is made up by defining the character as a series of eight bytes, each byte being one line of the character. A 1 indicates a pixel and a 0 the absence of one. Other new functions include READ, DATA and RESTORE, something which was sorely missing on the ZX-81. FN and DEF FN are also there.

One of the best new additions to the specification is the ability to type in lower-case. That certainly makes reading and writing programs easier, especially as the keywords are still capitals. So strings, variables and arrays can be specified in a way which is simpler to use.

Unlike the ZX-81, the Spectrum uses true ASCII codes for its alphanumeric characters and control codes. That means that ZX-81 cassettes cannot be read into the Spectrum. Other than the absence of SLOW, FAST and SCROLL, however, the Basic is identical. There is no need for FAST or SLOW, as the memory-mapped screen overcomes the need for screen writing during interlacing, as on the ZX-81. The Spectrum has the speed of the ZX-81 in fast mode with the screen characteristics of the ZX-81 in slow mode.

The Spectrum has an expansion port similar to the ZX-81, with the addition of the colour video information. Thus a colour monitor could be attached to give a high-quality display. Full data, address and control buses for the Z-80 processor are available and the ZX printer can be plugged-in directly.

The LPRINT, LLIST and COPY commands work with Spectrum Basic, with the additional bonus that any user-defined graphics will also be printed. It is also possible to run a number of other peripherals at which Sinclair has only hinted. There will be an RS232C interface, so that standard printers can be attached. There will also be a network with an interface which fits on the expansion port, as will the ZX-Microdrive, to be launched later. It is possible to access all I/O ports by using the IN and OUT commands in the Basic.

The Spectrum has a very basic sound capability. The internal speaker emits a 'raspberry'-like sound, set normally at a frequency of middle C. The pitch and duration of the note can be defined in the Basic with the BEEP command. The centre frequency being middle C, any other note can be defined by the number of semitones above or below that frequency. It is also possible to have fractional intervals.

In summary, the ZX Spectrum is a very fine computer and at the price will give Commodore, Acorn/BBC and Atari a run for their money.

Taking the lid off

Stephen Adams

THE SPECTRUM is very similar in shape and style to the ZX-81 and there are many similarities. Sinclair retained the original keyboard to save space but has provided a rubber sheet with moulded keys on it which fits over the top. The sheet is suspended over the flat keys — which on the Spectrum are bigger than the ZX-81 — so that when a key is pressed it bends to give some feel to the keyboard.

That, and the fact that the keys repeat if held down for longer than one second, even when using SHIFT, makes the keyboard much easier to use. The single keyword system has been retained and that saves memory, as all the Basic words can be stored as one byte. It also means that two SHIFT keys are required to reach all the functions; one is called CAPS SHIFT and the other SYMBOL SHIFT. They are at opposite ends of the keyboard and as they are often used one after another, it tends to slow the input speed as you are constantly swapping hands.

For instance, RUBOUT and the cursor movements use CAPS SHIFT and +-* are SYMBOL SHIFT. It would have been a better idea to put both on the left-hand side, as they often need to be used together and could be pressed with one hand while the other searches for the appropriate key.

The keyboard is an input-output mapped device, as on the ZX-81, and

along with the ZX printer, which is the same for the ZX-81 and the Spectrum network RS232 interface, discs, loudspeaker, tape interface and border colours require only one address line to work. That means that you must make all of the lower five address lines a binary 1 to use your own devices.

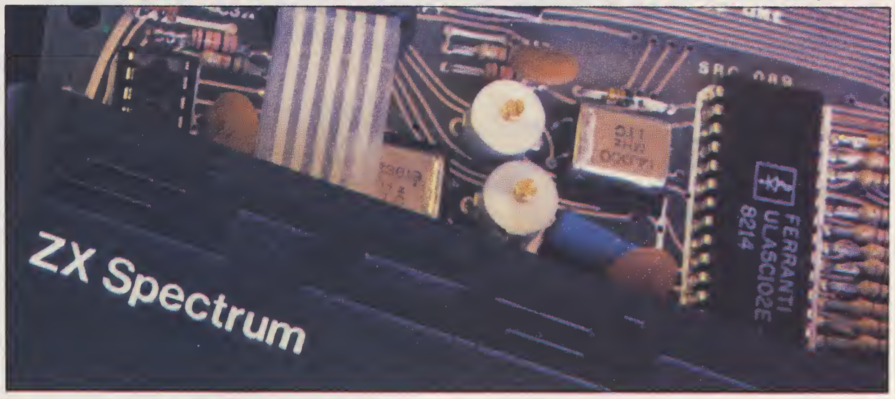
The input-output map access has been improved greatly, however, with the addition to the Basic commands of IN and OUT. They give an instruction IN A(c) or OUT A(c) where registers BC give an address from 0 to 65535.

The memory-mapped addressing of the RAM/ROM occupies 0-16K and the RAM 16K-32K on the basic 16K model. There is provision for an extra 32K board to be plugged into IC sockets at the back of the printed circuit board. The 48K version will have the board fitted but to add it later will cost £60, which seems expensive.

There would be no difficulty in adding extra ports to the memory map, as on the ZX-81, above 32K — on the basic version — but for two things. There is no line, so that the extra RAM can be turned-off if required on the edge connector and the edge connector address lines have been moved to the outer edges so that it is incompatible with the ZX-81. The Spectrum has a 28-way double-sided edge connector of the same style as the ZX-81, with the

keyway on pin 5. That makes any input-output device compatible with the ZX-81 but any memory-mapped devices would have to be re-arranged. The edge connector also has a number of new signals on it which are not explained in the manual, plus a video output and colour outputs for VDUs.

All the voltages used on the Spectrum are also brought out, namely +5V, -5V, +12V and -12V. They are obtained from the same buzzing transformer as is used in the 16K RAM pack and most of that



RAM pack seems to have been transplanted on to the Spectrum.

The obvious additions to the circuitry are the PAL colour mixer under the metal can, which contains the video modulator, and two crystal-controlled oscillators are used, one for the ULA, which controls the screen among other things, and the other for the colour mixer. The 14mHz clock for the ULA is also used to drive the Z-80A microprocessor after it has been reduced to 3.5mHz. That is 0.25mHz faster than the ZX-81. The Z-80A has also been freed of the job of putting-out the screen — by the ULA — and so no longer requires the commands FAST or SLOW, as it works at top speed all the time except when BEEP or PAUSE is used.

PAUSE and BEEP both cause the Z-80A to stop for a time determined by the programmer and so it will do nothing else while those commands are being done. BEEP commands should be kept short in a program for that reason; 0.01 seconds is a good speed to PRINT AT and BEEP at the same time.

As for programming the Spectrum, it can be considered as an extension of the ZX-81 Basic. The PAPER, INK, BRIGHT and FLASH commands for each character square are stored in a memory map above the dots for each character. They are all stored in one byte per character and can be read by the Basic word ATTR and altered either by Basic commands or POKEs.

The dot screen is a different matter, however, and cannot be altered so easily, as the dots are stored in peculiar order, so you have to use the graphics commands — which can define all the dots on the 22 line by 32 character screen available to the user — or the SCREEN command.

The screen takes up 6,912 bytes of the 16K memory and the system variables take up another 738 bytes. The rest of the memory is not free for the user to use as 11 other areas float above location 23733 and can expand and contract as required by the Spectrum.

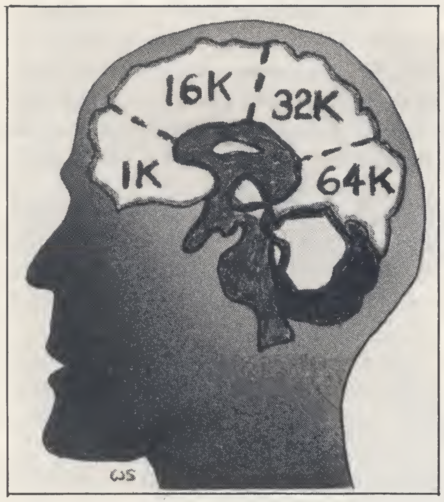
The program and variables are sandwiched in the middle of those, so REM statements cannot be used for machine code. There is an area, however, which can be used for machine code programming above RAMTOP which is ignored by the Basic and its length can be defined by the user.

The user-definable characters area is stored above that so they can be retained from program to program.

There are many tape arrangements which can be made with the Spectrum. The program, strings or machine code can all be SAVEd, LOADed and VERIFYd separately. The variables and screen can also be stored on tape but cannot be VERIFYd. As each is SAVEd, a message to start the tape recorder will appear and wait for you to press a key.

SAVEing or LOADing causes the border to flash red and green or red and blue, depending at which part of the tape you are looking. All the tape programs LOADed correctly.

MISCELLANY



During the year *Sinclair User* has published a number of articles of a specialist interest. We reprint some of them, particularly from the early editions which are now unavailable, together with some new features on issues in the world of Sinclair computers. We begin with Ian Logan, who has some interesting comments on the ROMs of the ZX-80, ZX-81 and the Spectrum.

Ian Logan looks inside the ROMs

Monitoring the success of Sinclair

THE NUMBER of Sinclair computers manufactured is now approaching 750,000. That impressive figure shows the wide acceptance of the Sinclair design and Sinclair Basic. So far there have been three monitor programs — the 4K program of the ZX-80 and the 8K program of the ZX-80 and ZX-81; and now the 16K program of the Spectrum — and each has achieved a success which should be the envy of any machine code programmer.

It was clearly one of Clive Sinclair's earliest decisions to have a radically new Basic interpreter written for the ZX-80. It would have been easier to use an existing interpreter, e.g., from Microsoft, but it is questionable whether such an interpreter, the operating system and a character set could have been fitted into a 32K-bit ROM 4K-byte read-only memory.

So the 4K program was commissioned and written. The brief for the programmers clearly was to write as full a Basic interpreter as was possible and to include short but still functional routines for scanning the keyboard, producing the display and handling the cassette interface.

The 4K program was a great success, despite the fact that it handled integer-only arithmetic. The full use of single-stroke keywords was completely successful and the impressive syntax-checking feature made the ZX-80 an instant success. The 4K program was written *de novo* and was very structured. It was also almost totally bug-free.

The 8K program was written for the ZX-80 and re-written to run the ZX-81. The ZX-80 was almost out-of-date at the time of its launch and within a year Sinclair had the ZX-81 almost in working order. The ZX-81 was developed directly from the ZX-80 but there was never a possibility that the SLOW-FAST machine code handler could have been incorporated into the 4K program.

In any case, prices of chips were falling fast and the ZX-81 was always to be supplied with a 64K-bit ROM 8K-byte read-only memory.

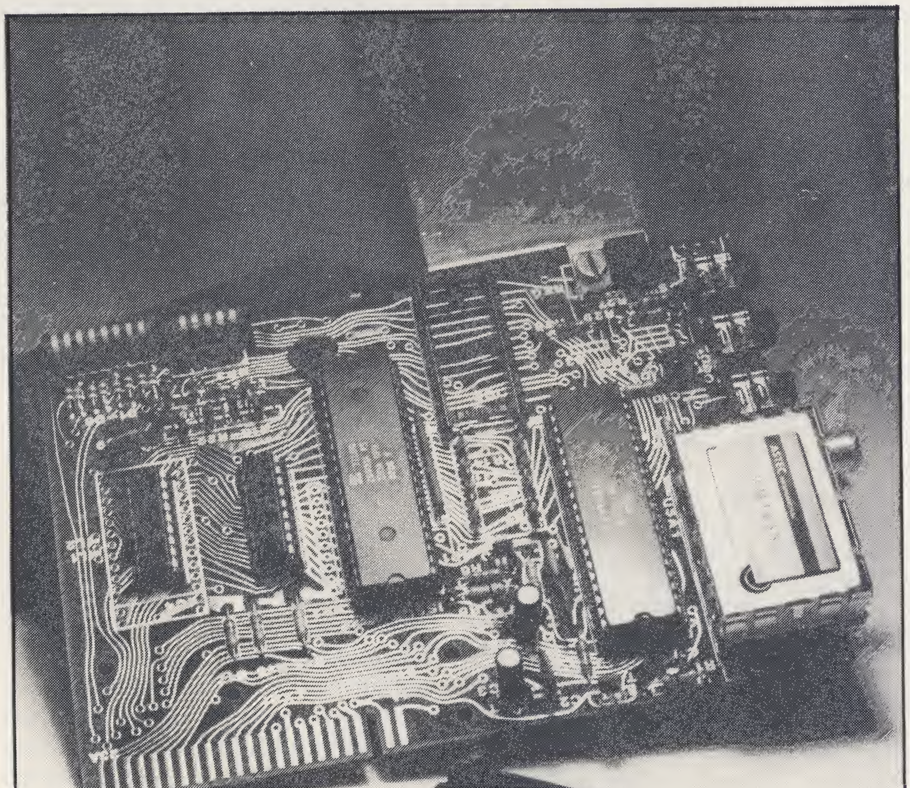
The 8K program was written by taking the old 4K program and adapting it. Much greater emphasis was given to the tape interface routines — but still not sufficient; the SLOW-FAST handlers were added and the integer calculator replaced by a very sophisticated floating-point calculator which used an internal stack-operating language.

Once again, there was insufficient room available and the programmers had to omit routines they would have liked to include and to write economical code, thereby sacrificing speed.

Moreover, the whole program clearly was written in a hurry. The initial well-structured approach used in the 4K program was lost and the result unfortunately was not too successful. All in all, there were probably about 100,000 ZX-81s supplied with the unimproved 8K ROM program. Those machines had a fault in the SLOW-FAST handlers which made the machine crash unless the user included POKE 16437,255 at suitable places in a Basic program and, more important, the machines could not count correctly, e.g., PRINT 0.25**2 gave 3.1423844.

Whereas the error in the SLOW-FAST handler could be corrected by software, there was no such possibility of correcting the calculator error in that way. It was possible, however, by means of a hardware add-on in effect to change the code at location hex 1735 and thereby correct the error. It is probable that about 20,000 ZX-81s were fitted with that hardware modification in the summer of 1981.

Faced with the somewhat unsatisfactory situation, Sinclair had the 8K program re-written and machines with the improved pro-



gram began to appear in September, 1981. Since then, the 8K program has been bought, along with a ZX-81, by about 400,000 customers and that surely makes it the most successful machine code program ever written.

Notwithstanding its undoubted success, it is interesting that even now it is not bug-free. The most important error occurs in the 'division routine' e.g.:

```
10 LET A = 0.00001
20 LET B = 1/10000
30 IF A«»B THEN PRINT A;"
   «» ";B
```

which is well within the capabilities of four-byte mantissa arithmetic which should give about eight decimal figures of accuracy. To the user the error appears as a rounding error and shows-up only occasionally. It is, though, definitely a bug, as it was not the programmers' intention to have that kind of result.

In April, 1982 the Spectrum was launched in a blaze of publicity and it is a most worthy successor to the ZX-81. Now, at last, there is a satisfactory set of cassette-handling routines; a high-definition display which allows for the drawing of lines, arcs and circles; a small beeper, and, of course, satisfactory colour.

The situation of having a high-resolution display for handling points but a low-resolution format for colour is clearly a hardware compromise and the machine would never have been so cheap if it were to include an additional set of display chips.

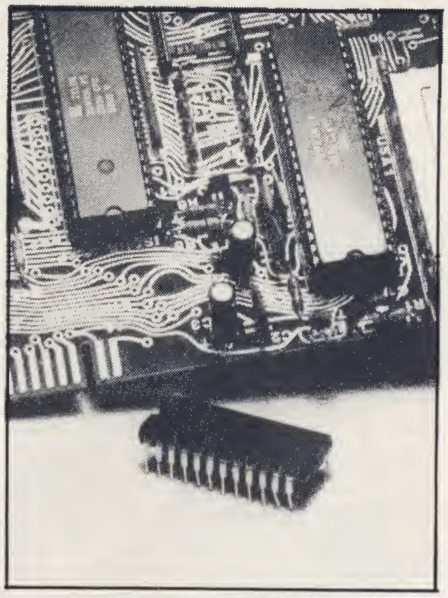
The Spectrum is a direct development from the ZX-81 in respect of its hardware and its monitor program. The 16K program is an adapted 8K program and contains large blocks of code which have been carried, unchanged, from the earlier program to the later one. To those blocks have been added routines for all the new commands.

Essentially, therefore, the pedigree of the 16K program can be traced to the 8K program and then further back to the original 4K program. The tidiness of the first program is now almost completely lost and the present 16K program has only a token attempt at structure.

It is probably a direct consequence of the loss of the overall structure of the program that it contains a large number of bugs. In addition to the division error carried-over from the 8K program, there are errors in the routines for back-spacing, forward-spacing, SCREEN, CLOSE and STR$. Also there has been considerable confusion with respect to the value '-65,536'; leading to e.g., PRINT INT -65536 giving -1.

It is to be hoped that sooner rather than later an improved version of the 16K program will appear.

Has Sinclair given a new lease of life to a failing language with its Basic? It is very interesting to see how many people speak well of Basic, as opposed to its opponents



who seem to be so outspoken about their feelings against it.

Basic is an almost perfect language for beginners; it is easy to start and is sufficiently like English to make sense. It does not lend itself well to the writing of highly-structured programs but it is always surprising to see just how well-formed a good Basic program can be.

The fact that Clive Sinclair chose Basic for the ZX-80 and has continued to use it in the ZX-81 and the Spectrum shows that at present there is really no worthwhile alternative.

The success of Sinclair Basic is due to its overall clarity of presentation as much as to any of its special command features. The design of the characters is clear and bold, and the use of both leading and trailing spaces which appear automatically is very impressive. With respect to the command features the use of

- PRINT AT line, column; -

is extremely easy to use and a great advance on the use of a series of cursor home, cursor down and cursor right control characters; but it would have been pleasant also to have the use of a PRINT character area command. The provision of allowing for the length of a string to be fully-defined is also of great advantage and that, together with the slicing feature, makes the manipulation of strings straightforward.

In no way can it be said that any of the three Sinclair machines is supplied with Extended Basic but within the constraints of the size of the ROM allowed in the particular model, the programmers have produced three levels of excellent Basic.

Two more special features have been found in all the Sinclair machines. Automatic syntax checking of the edit-line has been a feature in all three Sinclair models and there is probably no other single feature which makes for better programming. Immediately an error has been made the machine prompts the user to correct it.

In most machines it is only run-time errors which are detected but in the Sinclair machines the Basic interpreter is called to check the syntax of Basic lines prior to its being called to run the lines.

The two passes through the Basic interpreter are distinguished from each other by the use of syntax/run flag bit 7 of the system variable FLAGS. During syntax checking, the flag is re-set and at any stage when a value normally would be changed, the flag is read and, in consequence, the value is left unaltered. Later when the line is being run the flag is set and that leads to the changing of values.

The same error reports are generated whether in syntax-time or run-time but it is a feature of the operating system that the user is not given the report when the syntax fails; only an error marker appears in the edit-line and then only at the position reached by the interpreter when the error occurred.

Fundamental to all the Sinclair machines has been the use of single keystrokes for Basic command words, functions and some separators. That feature has been a great success, although on the Spectrum

the number of different keystrokes possible is leading to an interesting range of styles for entering programs — it is scarcely touch typing. How does the machine know whether to write a K or L, or for that matter on the Spectrum a C, E or G at the correct moment?

It is interesting to find that the K/L cursor is produced, not as a result of scanning the line, but as a result of trying to print the line.

When an edit-line is apparently being written on the lower part of the screen it is in reality being built-up in the editing area and then printed, with tokens expanded, in the appropriate part of the display file. It is that printing which leads to the cursor being K or L. The first character after a line number is printed in K-mode; thereafter the machine is in L-mode unless a THEN is used or a new statement begun.

Then as the user enters a keystroke the machine considers its current mode when determining which value to return for that key.

In the Spectrum, C-mode — caps lock — is distinguished from L-mode by the flag — bit 3 of FLAGS2; and E-mode and G-mode by bits 0 and 1 of the system variable mode.

As machines for learning about computing, playing games and solving problems the Sinclair computers are clearly in a class of their own; but their fragility leads to their use being limited.

No doubt the price of both the ZX-81 with its fine black and white display, and the Spectrum with its beautiful colour display, will fall steadily over the next year.

It is to be hoped, however, that the monitor program of the Spectrum is re-written and that consideration is given to the production of a Spectrum Mark II which uses the same software, runs the same Basic and machine code programs, but produces a stronger colour signal.

For further details see *Understnding Your Spectrum* — by Dr Ian Logan *Sinclair ZX Spectrum ROM Disassembly* — by Dr Ian Logan and Dr Frank O'Hara.

The first book is due to be available in a few weeks but the 'disassembly' which will be fully documented, will probably become available only early 1983.

Dr Ian Logan won the 1981 Rosetta Award and is the author of a series of books about the ZX-80, ZX-81 and Spectrum.

ZX-80 programs can be converted

Mark Charlton gives a step-by-step guide

LOOK QUICKLY through the listing of a program you want to convert. If there are no PEEKs and POKEs and you have more than 1K on your ZX-81, you are not likely to have many problems. The main difficulties in converting programs occur when there is POKEing to the screen, or when the ZX-80 program proves to be too long for the 1K ZX-81 when converted.

Despite that, most programs will run more smoothly on the ZX-81, PRINT AT and SLOW ensure that far more attractive displays are possible and the use of inverse graphics in PRINT lines can also enhance the display using INKEY$ instead of INPUT A$; pressing NEWLINE also makes things work in a much more satisfactory way.

If you look through the program before you enter it and see there are no PEEKs and POKEs, you should make a few changes on the listing before turning-on your computer. Change the way the random numbers are generated, for a start:

```
ZX-80
LET X = RND(10)

ZX-80 LET X = INT(RND*10) + 1
```

You may be able to save a line when you see the following in a ZX-80 program:

```
LET X = RND(10)
IF X = 5 THEN PRINT "FIVE"
```

The ZX-81 version could well be IF RND greater than or equal to .9 THEN PRINT "FIVE".

After changing the random numbers — and realising, of course, that RAND on the ZX-81 is equal to the ZX-80 RANDOMISE — look for any use of TLS (truncate left) in the ZX-80 program.

The ZX-81 equivalent of this very useful ZX command is A$ (2 TO). That is, if the ZX-80 program says:

```
LET A$ = "HELLO"
10 LET A$ = "HELLO"
20 Print A$
30 LET A$ = TLS(A$)
40 IF A$ greater than "THEN
   GO TO 20"
50 PRINT "GOODBYE"
```

You should rewrite it to read:

```
10 LET A$ = "HELLO"
20 PRINT A$
30 LET A$ = (2 TO)
40 IF A$ greater than "THEN GO
   TO 20"
50 PRINT GOODBYE
```

In both cases, the output of the program will be:

```
HELLO
ELLO
LLO
LO
O
GOODBYE
```

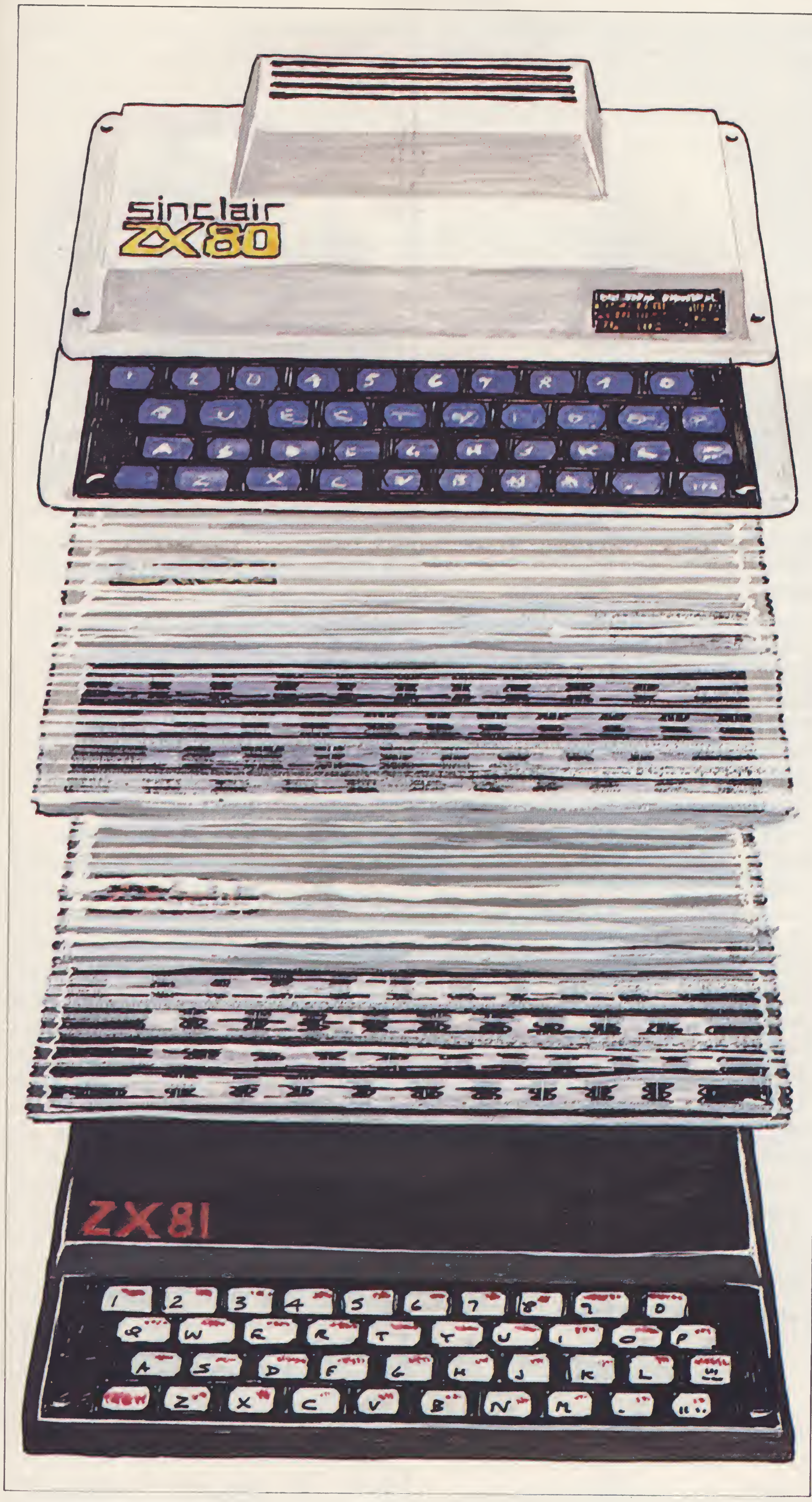
The ZX-80 lacks the facility to deal with floating point arithmetic, so it truncates a number automatically. The INT function should be used before any division in a ZX-80 program listing when entering it into the ZX-81. You may find that you would prefer the computer to round the figure to the nearest whole number, rather than whole number:

Here is a small program on the ZX-81 to show that the INT function does not 'round' to the nearest number but just to the nearest lower whole number:

```
     10 INPUT A
     20 LPRINT A, INT A
     30 GOTO 10
1                          1
1.4                        1
1.7                        1
1.99                       1
```

The output shows the numbers entered (1, 1.4, 1.7 and 1.99) and what happened to them. You need to add .5 to the numbers entered if you want rounding-up, as the following example shows, when entering the same initial four values:

```
     10 INPUT A
```




```
20 LPRINT A,
30 LPRINT INT (A + .5)
40 GOTO 10
1                  1
1.4                1
1.7                2
1.99               2
```

Certain functions on the ZX-81 — PRINT AT, PLOT and TAB — 'INT' number automatically, so work on which value — the nearest whole number or the lower whole number — to which you would like them be INTed.

If you have a program which stores information in and retrieves information from REM statements you can convert them easily by keeping in mind that the first address after the word REM in the first line of a ZX-80 program is 16427, while the equivalent address on the ZX-81 is 16514. Here is a list, produced by the hard-working ZX-81, of ZX-80 (first) addresses for PEEK/POKE REM work and the equivalent on the ZX-81:

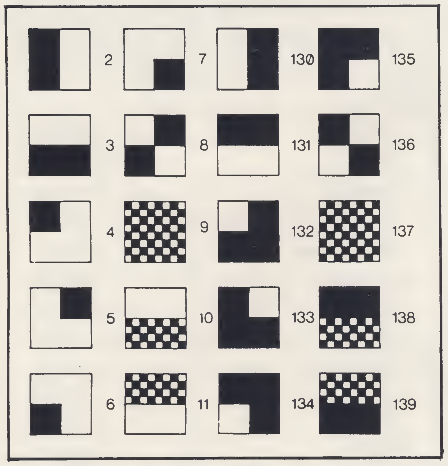
ZX80		ZX81
16426	< - >	16513
16427	< - >	16514
16428	< - >	16515
16429	< - >	16516
16430	< - >	16517
16431	< - >	16518
16432	< - >	16519
16433	< - >	16520
16434	< - >	16521
16435	< - >	16522
16436	< - >	16523
16437	< - >	16524
16438	< - >	16525
16439	< - >	16526
16440	< - >	16527
16441	< - >	16528
16442	< - >	16529
16443	< - >	16530
16444	< - >	16531
16445	< - >	16532
16446	< - >	16533
16447	< - >	16534
16448	< - >	16535
16449	< - >	16536
16450	< - >	16537
16451	< - >	16538
16452	< - >	16539
16453	< - >	16540
16454	< - >	16541
16455	< - >	16542
16456	< - >	16543
16457	< - >	16544

You are probably aware that all graphics symbols, letters and numbers, and their inverses, are available directly from the keyboard on the ZX-81. That was not so on the ZX-80 and, depending on the listing, you will find either a great number of things like PRINT CHR$ (128) to print an inverse space, or that the listing draws-in the required character or refers to the graphic on a particular key.

If the specific key is referred to, use this table to convert, keeping in mind that the first reference is to the ZX-80 keyboard, the second to the ZX-81: shift Q, graphic 5; shift W, graphic 6; shift E, graphic 1; shift R, graphic 2; shift T, graphic D; shift A, graphic A; shift S, graphic T; shift D,

graphic 4; shift F, graphic 3; shift 6, graphic S.

Here are the graphics symbols available on the ZX-80, with their character numbers and codes. Use this table when you find a reference



to, say, PRINT CHR$ (137) in a ZX-80 listing:

You may find that the appearance of the program when running will be greatly improved if you use PRINT AT rather than CLS. Experiment with display formats to see if you can dispense with use of CLS. The SCROLL command is very useful for printout of tables.

ZX-80 programs which use a moving display routine are generally greatly improved on the ZX-81. You will find it necessary to delete the whole of the moving display, changing it to PAUSE — jerky, but the only thing possible on a new-ROM ZX-80, unless you have added a SLOW — or to a delay loop — FOR J = 1 TO 20 NEXT J — or just leaving it as it is. You can obtain some indication as to which figure to place after the word PAUSE by keeping in mind that the figure used in most ZX-80 moving-display routines to POKE 16414 is related inversely to the delay — that is, the larger the number, up to 255, the shorter the time the display is held on the ZX-80.

Many ZX-80 programs use the following line to POKE a character into a specific position on the screen: POKE Y*33 + X + 1 + PEEK (16396) + PEEK (16397)*256,n

Wonder of wonders, this still works on the ZX-81 adjusting itself automatically to accommodate changes in programs. Y is the count-down from the top of the screen, X is the count across. You may well prefer to use PRINT AT, which produces more or less the same result.

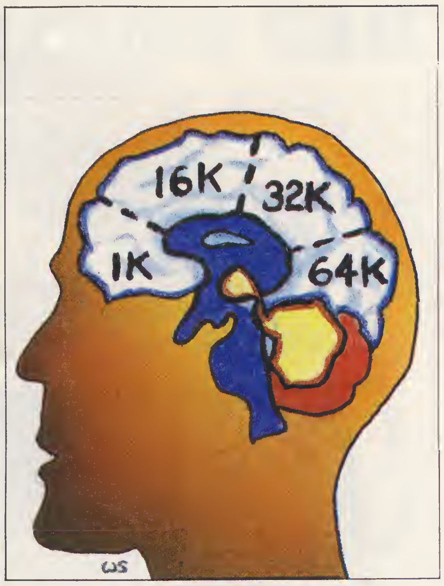
Not all system variables are so convenient. Here is a list of them, showing the ZX-80 and ZX-81 equivalents. The list was compiled by Toni Baker:

Old ROM	New ROM
16384	16384
16385	16385
16386	16391
16387	16392
16388	no equivalent
16389	no equivalent
16390	16394
16391	16395
16392	16400
16393	16401
16394	16404
16395	16405
16396	16396
16397	16397
16398	no equivalent
16399	no equivalent
16400	16412
16401	16413
16402	16418
16403	16419
16404	16420
16405	16408
16406	16409
16407	16427
16408	16428
16409	16429
16410	16432
16411	16433
16412	16434
16413	16435
16414	16436
16415	16437
16416	no equivalent
16417	no equivalent
16418	no equivalent
16419	no equivalent
16420	no equivalent
16421	no equivalent
16422	16406
16423	16407
16424	16509
16425	16510
16426	16513
16427	16514

To sum up, here is what you do:

ZX-80	ZX-81
LET A = RND(9)	LET A = INT(RND*9) + 1
LET A$ = TL$(A$)	LET A$ = A$(2 TO)
Moving graphics: POKE 1641,n	PAUSE 255 — n (very approximate)
Variables.	see table
Graphics.	see relevant table, two given
LET A = C/B	LET A = INT(C/B) or LET A = INT(C/B + 5)
INPUT A	LET A = CODE (INKEY$) — 28

Good converting.



Memory explored

Flexible spaces

WHEN Clive Sinclair designed his basic computer, it was to contain only three things within the memory map, the area where all the instructions to operate the computer are stored. The area is easily accessible by the Z-80A microprocessor and it is easy to program uses for it. The memory area is divided into 65535 locations (64K), by the 16 binary address lines (AO-A15) which are used to indicate the location at which the processor wants to look.

Sinclair required only a place to store a program (RAM), a place to hold instructions to operate the computer when it was first switched on (ROM) and a location in memory which would translate the TV picture. So it divided the memory map into three sections, O-16K for the ROM, 16K — 32K for the program RAM and 32K — 64K for the TV picture. That was done by using only the top two address lines, A14 and A15, to tell which section was operating at the time.

This divided the memory map into the four sections as shown in figure one with the ROM and RAM repeating itself again and again. The address line A14 is used to turn on the ROM when the condition is binary O and when it is binary 1 the

RAM is turned on. The A15 address line is used to operate the hardware which puts out the TV picture; it operates only when A15 is binary 1.

As A14 can also change state from 0 to 1, while A15 is operating the TV hardware the RAM can be used to store the screen display. The software in the ROM makes use of that fact and has only to pick up the address where the screen is — between 16K and 32K — and then add 32K to it to put it on the TV screen.

The only problem with this system is that ROM between 32K and 48K cannot be used, as it would upset the TV picture. It also meant that the ROM appeared in the 8K to 16K section, blocking-out any other use for it, as it could not be turned off.

That was the situation which existed in the ZX-80 and only internal modifications to the computer circuitry could sort it out. The only place left to put new devices produced by other manufacturers was to steal some of the RAM space to fit in the new device. That was done by turning-off the RAM, operating a pin on the expansion port called RAM-CS. That was included in the design, as some method was needed to turn off the 1K of internal RAM when the 16K RAM pack was fitted on the back.

That was because the 1K RAM was repeated through the whole of the 16K section and would get in the way when 16 separate 1Ks were installed instead. When the 16K Sinclair RAM was installed that was not possible, as it not only blocked-off any connection to the expansion port by covering it but also used the only free memory locations. It also did all of its address decoding inside the pack, so could not be turned off by operation the RAMCS line.

When Sinclair saw that there was a market for his computer and that other manufacturers wanted to use it as a basis for experimenting with computers, he re-designed it, giving greater flexibility to the memory map. The result was the ZX-81.

The internal design was much the same as the ZX-80 — figure one — but this time an extra connection in the expansion port had been put to good use. It was called the ROMCS pin and by operating it in the same way as the RAMCS pin, the RAM could be taken out of the memory map. This was located on pin 23B on the expansion port and caused some chaos at first, as it was found that the first Sinclair attempt at more

Sinclair's Intended Memory Map.

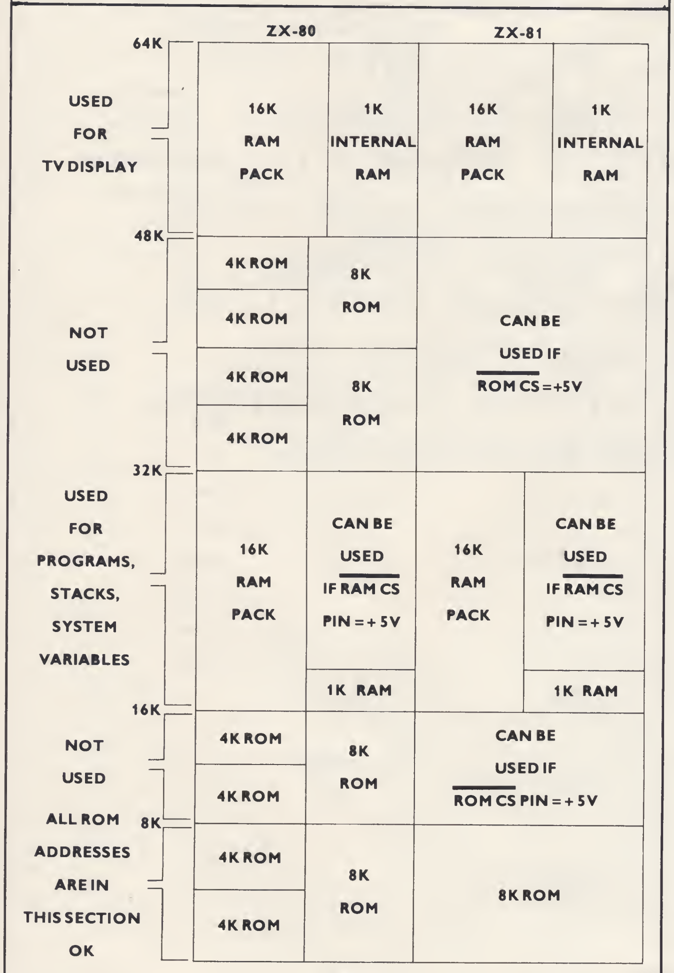


Figure 1

Memory Map O-16K

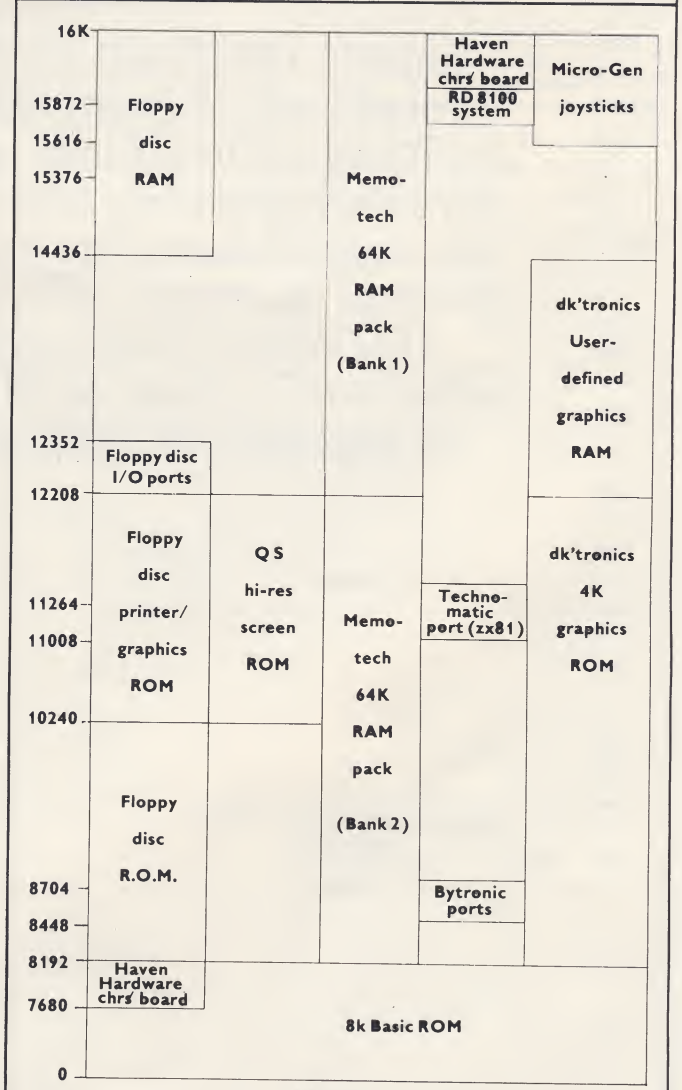


Figure 2.

external memory, the 3K RAM pack, did not work the ZX-81. This was because inside the pack the pin was kept at a permanent binary O and prevented the ROM from being turned on. The only solution to the problem was to cut the connection to the pin within the pack, leaving it unconnected to anything within the pack.

People could then, from the outside of the computer, free some space in the memory map for their use. That was a great advantage, as many of the people who had bought the ZX-81 did not want to have to fiddle inside their precious computers with a soldering iron for fear of damaging them.

It left many firms which had cut their teeth on the ZX-80 with a vast market for adding additional boards to the ZX-81, which could greatly expand its flexibility and use. Many of the people who started in the field have now left their original jobs to work full-time producing add-on boards for the ZX machines.

It also led every manufacture to try to use the same areas for different uses. Also because the ZX customer was not willing to pay a great deal for any extras to his machine, costs had to be reduced to fit the market. So they took a tip from Sinclair and reduced the amount of lines they used for addressing.

That meant that we had the same problem as previously, that one port covered more than one memory location; some cover as much as 8K. That, of course, means that when one device is using that area, no other devices can be used at the same address.

Figures two, three and four show as many of the devices which could be found, plus their addresses.

They also show, by putting into different columns, the devices which cannot be used together.

Included are two RAM expansions, as they all have to start at 16K and work their way upwards. That memory must be continuous for Basic, as the program would crash if it tried to store memory in a RAM location which was not there.

This is the first known attempt to try to catalogue all the devices the ZX-80 and ZX-81 can use . The only one lacking the manufacturer's name is the floppy disc system which is manufactured by Macronics.

For more details on these devices, contact the manufacturer.

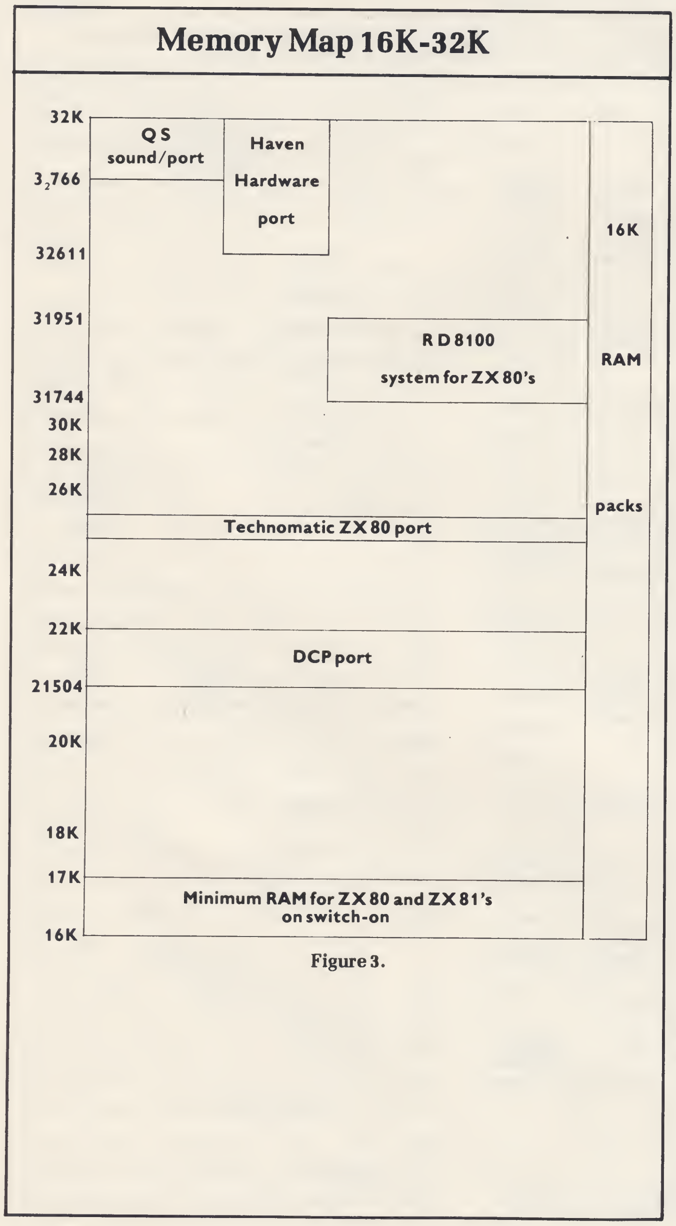


Figure 3.

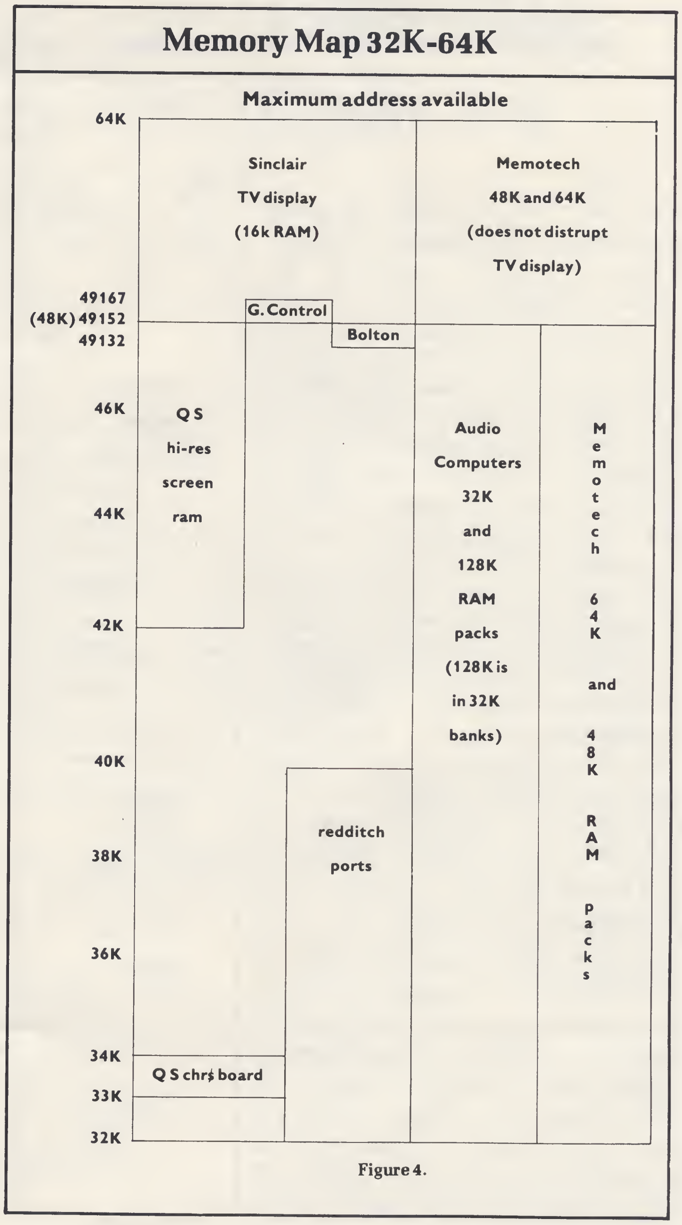
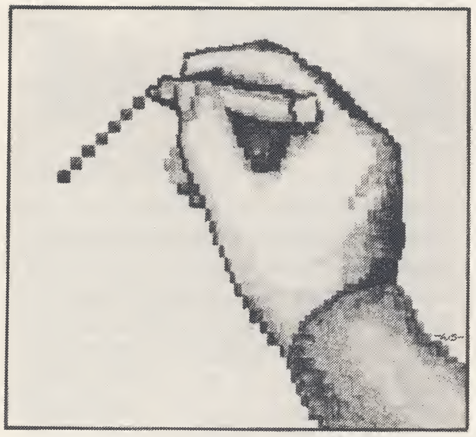


Figure 4.

GRAPHICS



One of the areas in which the ZX-81 cannot compete with other machines in the home computer market is in the quality of its graphics. With a little imagination and patience, however, it is possible to produce some good designs, as Phil Garrett discovered. The Spectrum arrived with better graphics but there were still improvements which could be made. Christopher Leigh explains the method for doing it.

Phil Garrett on the ZX-81's abilities

Making programs more attractive

GRAPHICS on microcomputers are advertised in much the same way as soap powder. Even the smallest soap packet is Giant Size and graphics start at high resolution and go on to ultra-high resolution.

The ZX-81 has high-resolution graphics.

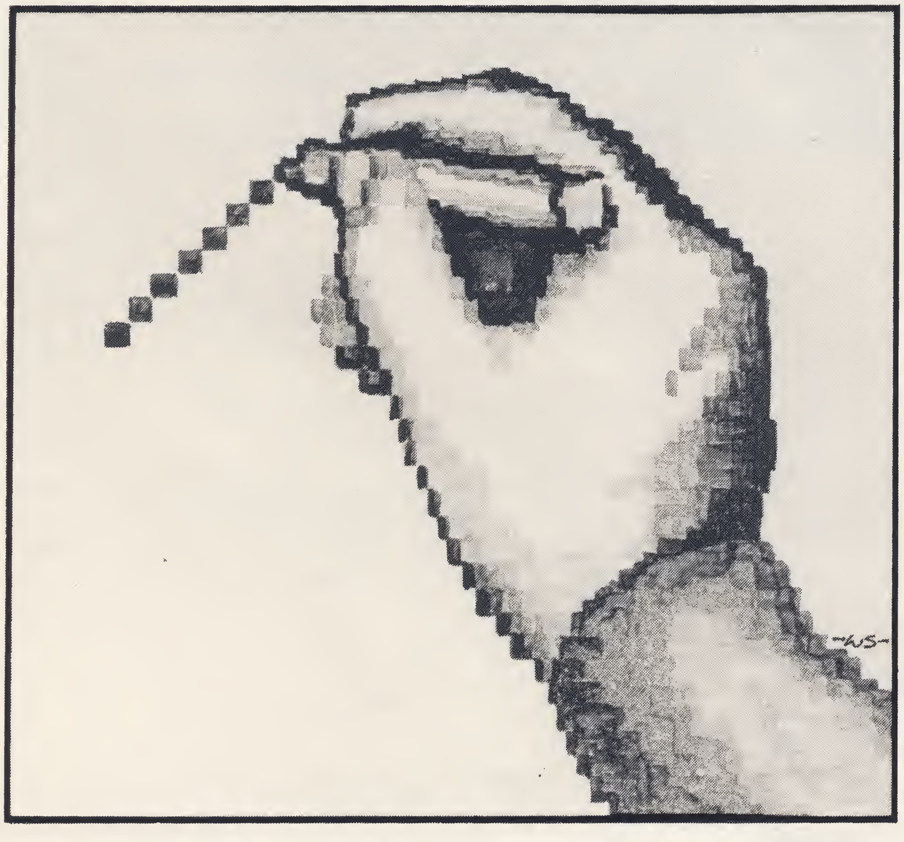
The extensive use of visual display units is a comparatively recent development. Before their spread, people were perfectly happy to use a computer with no graphics, even for games.

There are still some who regard graphics as a distraction but next time you meet a professional computer worker, ask him or her to recommend a book to explain computers to the absolute beginner. Nine times out of ten they will recommend the "children's" Ladybird book, *The Computer*. One of the reasons is that each page of text is accompanied by a clear and relevant illustration.

In the same way computer graphics can help keep attention in an educational program, or can present information in a more easily-assimilated way using charts and graphs, or can just make any program more attractive and interesting.

The ZX-81 high-resolution graphics have a definition of 64 pixels (picture elements) across by 44 pixels down, which is fairly low as high resolution goes. The Atom offers 256 × 192, the BBC model B 640 × 256, although a standard Apple has only 40 × 48. The problem with the higher resolution is that much larger amounts of memory are required, 20K on the BBC machine.

Worthwhile results can be obtained with Sinclair-sized pixels. Recognisable maps can be drawn as in the Video Software educational Video Map passages, and turnings can be shown as in d'Ktronics 3D Labyrinth; the Sinclair manual gives examples of plotting mathematical functions and straight lines but the best on the market, with its imaginative use of the full character set, including pixels, is the J K Greye 3D Monster Maze.



Even the instructions are livened-up, with a clown doffing his hat to the player, or victim. The game, a machine code three-dimensional maze, contains the amazing monster, which, if you do not run away, gets larger and larger, lumbering towards you.

Producing good graphics is very rewarding but also extremely time-consuming. Fortunately there is some help available.

Butler, Currie and Hook's Print 'n Plot Jotter is an amazingly simple idea and a genuinely useful aid. The jotter is a pad of 100 A4 sheets printed with a separate numbered grids for ZX-81 printing and plotting, so you can sketch-out and amend your desired graphics without either computer or temper being overheated. When the design is complete, you simply transpose it into your program.

As well as the jotter, there is also a matt polyester film available with the same grid pattern. That has the advantage of being re-usable, with care, provided a hard pencil is not used. It is also translucent, so you can trace directly over suitably-sized photos, drawings and maps. The only minor quibble is that it would have been advantageous if the grids had been super-imposed, as in the Sinclair manual.

It took about 30 minutes to do the picture of Fungus the Bogeyman, which I would not have attempted without the Print 'n Plot film.

The aptly-named Picturesque Screen Kit 1 is a completely different aid, in the form of a package of machine code sub-routines contained in a single REM statement less than 1K in size. The subroutines can be called from a Basic program with USR statements and they include flicker-free scrolling up, down, left and right, clearing or reversing part of the screen, or all of it, and drawing a border round a specified area.

There is also a very handy non-graphics routine which saves and loads program variables at double speed and so allows the exchange of data files between programs.

It is possible to fudge the ZX-81 into giving genuine high resolution.

The ZX-81 is a digital computer, so not surprisingly it is all done by numbers. Every character is made up from a grid of eight-by-eight dots. A space character has all the dots off (white), and an inverse space has them all on (black).

Starting at address 7680 in the ROM, each character in turn has eight consecutive bytes which determine its pattern. Each byte represents a single row of the character and if we convert that byte from decimal, 0 to 255, to binary, 0000 0000 to 1111 1111, the pattern of dots on or off is revealed.

The Character Table Printer program has been developed in an attempt to make all this clearer. It shows the address in the ROM, its decimal contents, the contents converted to binary, and then the binary converted into spaces and inverse spaces.

If the character set could be changed at will, we would be talking about genuine high resolution with 256 (32 × 8) dots across by 192 (24 x 8) dots down. You cannot POKE into the ROM but there are other ways. Sinclair very kindly provides one way in the booklet supplied with its printer, which allows high-resolution graphics to be output to the printer.

The LPRINT function works by reading a character from the printer buffer, address 16444 to 16476, finding the pattern of dots from the table in the RÓM, and then sending that pattern to the printer.

The Sinclair fudge involves moving down RAMTOP to leave a 256-byte space, then copying the LPRINT routine from the ROM into RAM, not the area above RAMTOP. This routine is then altered slightly, so that instead of looking for its character patterns starting at address 7680, it looks from address 32255 instead, the area above RAMTOP.

```
 10 FOR A=7680 TO 8192 STEP 8
 15 FOR C=0 TO 7
 20 LET D=PEEK (A+C)
 30 PRINT A+C;"  ";D
 40 FOR B=31 TO 24 STEP -1
 50 PRINT AT C,B-12;D-2*INT (D/
2)
 60 PRINT AT C,B;CHR$ ((D-2*INT
 (D/2))*128)
 70 LET D=INT (D/2)
 80 NEXT B
 90 NEXT C
100 PAUSE 100
110 POKE 16437,255
120 CLS
130 NEXT A

CHARACTER TABLE PRINTER

7944   0     00000000
7945   126   01111110
7946   64    01000000
7947   124   01111100
7948   2     00000010
7949   66    01000010
7950   60    00111100
7951   0     00000000

7984   0     00000000
7985   60    00111100
7986   66    01000010
7987   66    01000010
7988   126   01111110
7989   66    01000010
7990   66    01000010
7991   0     00000000
```

Character Table Printer.

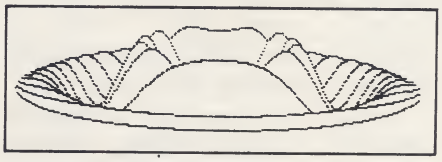
In those 256 bytes there is room for 32 characters, which happens to be the size of the printer buffer. We can then put whatever dots we want in the bytes above RAMTOP, fill the printer buffer with characters 0 to 31, call our special LPRINT routine, and they will be dumped on to the printer. Extra hardware is required if you want to have those high-resolution effects displayed on your TV.

The Quicksilva Character Generator is an add-on board which includes IK of extra RAM, addresses 33792 to 34815, which can be fitted with character patterns of your choice.

There is room for 128 characters, so you can have the Sinclair character set plus many more. Part of the hardware on the board detects when the ROM display routine is about to look up the pattern for a character in the ROM table and sends it to the patterns held in the extra RAM instead.

The Quicksilva Hi-Res Graphics Board interrupts the normal display routine in a similar way but is far more powerful and is a joy to use. The board has 6K of RAM, address 40960 to 47130, in which the high-resolution display is stored separately from the normal display file. It also has a 2K ROM, address 10240 to 12287, in the unused 8K between the Sinclair ROM and RAM.

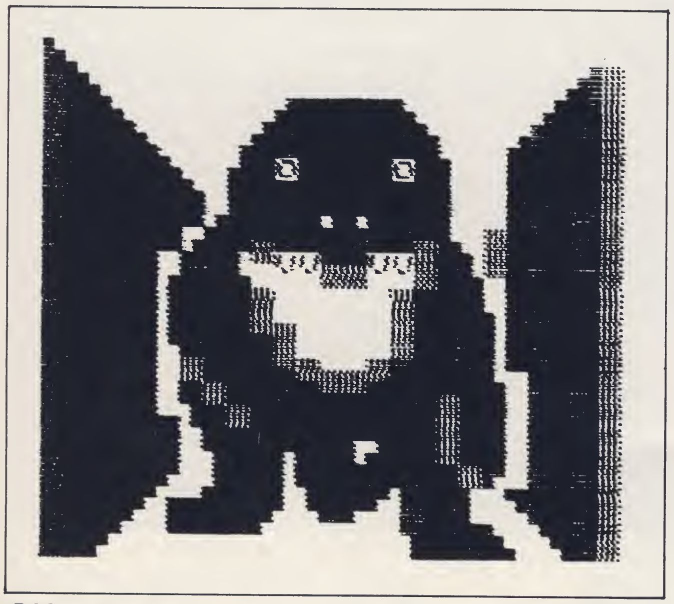
Routines in the 2K ROM can be called from Basic or machine code to perform high-resolution plotting



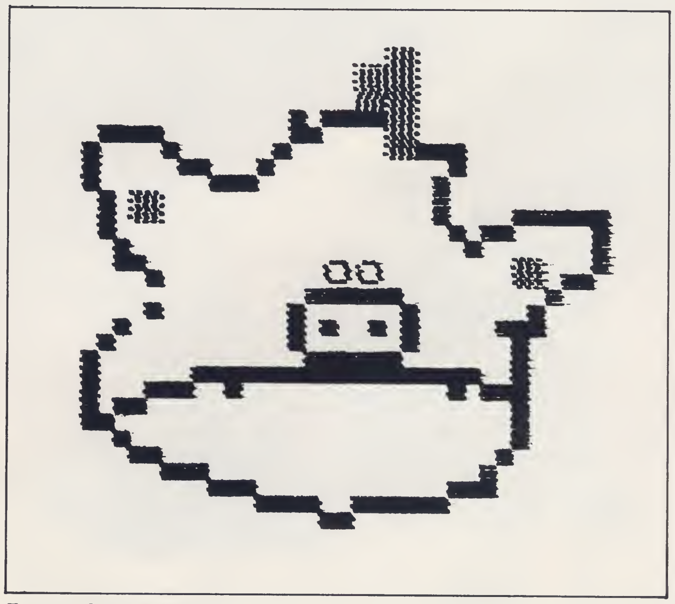
Quicksilva Hi-Res Graphics

and line drawing in black on white or vice versa. The hi-res board is expensive.

Whether you use the standard character set or hi-res add-ons, fascinating and worthwhile effects are possible on the ZX-81. A pictorial alphabet for teaching early reading would be very rewarding, though it would take a good deal of effort, and a very snappy flight simulator should be possible with a hi-res board.



3D Monster Maze.



Fungus the Bogeyman.

Christopher Leigh on the Spectrum

Improving on the manual

FIRST acquaintance with the Spectrum chess routine gave the impression that a magnifying glass was needed to identify the pieces.

The obvious aim was to make them bigger but that meant using more than one character space for each chess piece.

It would have been good to fill the screen with a 24*24 display but a 16*16 one was chosen. That meant that each chess-board square consisted of four printing spaces arranged 2*2.

Printing the board was fairly easy; the labelling was added later. A white screen was kept for ease of reading, though a black screen would probably look more impressive.

The board was printed using paper colour controls so that ink could be reserved for the pieces. It seemed more sensible than, for example, to print one large magenta and then to put green ink squares over it, which would have involved using Inverse, which can be confusing.

Using shaded graph paper, the pieces were drawn-in with a pencil for each little square — representing a pixel — which needed inking-in. That was translated into a binary code which could be POKEd. The user graphics are of memory.

Three considerations affected the shape of the pieces. They had to leave sufficient margin to show clearly the colour of the square on which they were printed.

They had to be very clearly distingishable. For that reason pawns were made smaller than other pieces. The third consideration was more awkward. Each piece occupies a 2*2 square, requiring four user graphics to define it. There are six different pieces in chess so that 24 graphics characters would seem necessary but only 21 are available.

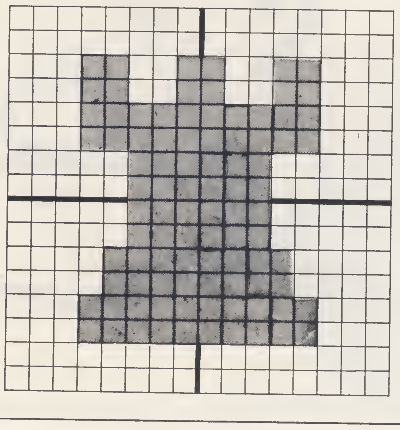
The answer was to share some of them between pieces; the fourth quarter of the queen is the same as the corresponding quarter of the knight and the complete bottom half of the king is the same as that of the rook, which consists of USR "a" followed by USR "b" with USR "e" and USR "d" underneath. Each drawing is labelled with the appropriate USR letter and then the binary codes worked out for each graphic. Each pixel is represented by a 0 for paper and a 1 for ink.

The codes can be POKEd into memory as binary numbers using BIN but as there are so many to do they can be translated into decimal to reduce the typing.

The original drawings and binary codes have been kept for ease of operation. Figure one shows how this works out for the rook.

Figure 1.

DECIMAL	BINARY
0	00000000
0	00000000
25	00011001
25	00011001
31	00011111
31	00011111
7	00000111
7	00000111
7	00000111
7	00000111
15	00001111
15	00001111
31	00011111
31	00011111
0	00000000
0	00000000



BINARY	DECIMAL
00000000	0
00000000	0
10011000	152
10011000	152
11111000	248
11111000	248
11100000	224
11100000	224
11100000	224
11100000	224
11110000	240
11110000	240
11111000	248
11111000	248
00000000	0
00000000	0

With the pieces complete the most laborious part of the job is over. The next task is to put the pieces on the blank board. The user graphics could be typed-in but that would make any kind of game impossible.

The pieces must be held in a string array. An 8*8*4 string array was chosen so that each board position was fully-defined and all four labels for the user graphics could be put into the array for each piece, which proves to be fairly simple. The use of upper- and lower-case letters gives the array an extra dimension, so the computer always knows the colour of any pieces referred to.

Once the board array is set up, printing it is merely a question of getting the numbers correct so that the proper graphic is printed at the correct position. Number juggling is also needed when entering moves.

At that stage the program can be run to see if it works. If the blank screen at the start of the program occurs while the machine processes lines 1-1000, to prevent the player thinking something is wrong lines 70-90 should be added.

Other improvements were made. There was a long wait between printing white and black pieces. The Spectrum prints blanks all over the centre of the board. Variable "w" is used to suppress such stupidity on the first run. On subsequent runs the player has to suffer space printing, as it is used to erase pieces which are out of place.

The complete listing is shown in figure two. The program puts board and pieces on the screen and then allows you to move them at will. Pawns automatically become queens on last rank; castling has to be done in two moves. Games can start with the pieces put in any position. Lines 70-90 print a title and flashing "thinking!" so the screen is not blank during the few seconds it takes to run through the user graphics and setting-up the board array.

Lines 100-350 poke the user graphics. Since only 21 are available, three are used twice by the chess pieces.

Lines 500-640 set up the two-dimensional string array which is used to hold the positions and the make-up of each piece. Each letter of the string points to the user graphic which defines a quarter of the piece. The use of upper- and

Figure 2.

```
70 BORDER 1: PAPER 1: INK 6: CLS
80 PRINT AT 9,6; BRIGHT 1;"SPECTRUM CHESS"
90 PRINT AT 12,10; BRIGHT 1; FLASH 1;"Thinking!"
95 REM
100 REM Set up user graphics
110 FOR n=1 TO 21: REM 21 user graphics
120 FOR m=0 TO 7: READ a
130 POKE USR CHR$ (n+143)+m,a
140 NEXT m: NEXT n
150 DATA 0,0,25,25,31,31,7,7: REM rook
160 DATA 0,0,152,152,248,248,224,224
170 DATA 7,7,15,15,31,31,0,0
180 DATA 224,224,240,240,248,248,0,0
190 DATA 0,0,4,3,3,5,15,31: REM knight
200 DATA 0,0,128,192,224,240,240,240
210 DATA 25,1,3,7,15,15,0,0
220 DATA 240,240,240,240,240,240,0,0
230 DATA 0,0,1,3,7,6,14,8: REM bishop
240 DATA 0,0,128,192,224,96,112,16
250 DATA 8,14,6,7,3,15,0,0
260 DATA 16,112,96,224,192,240,0,0
270 DATA 0,0,25,25,6,6,1,1: REM queen
280 DATA 0,0,152,152,96,96,128,128
290 DATA 15,15,15,15,15,15,0,0
300 DATA 0,0,1,1,7,7,1,1: REM king
310 DATA 0,0,128,128,224,224,128,128
320 DATA 0,0,0,1,1,7,7,7: REM pawn
330 DATA 0,0,0,128,128,224,224,224
340 DATA 7,1,1,15,15,0,0,0
350 DATA 224,128,128,240,240,0,0,0
360 LET w=1: REM initial position
500 REM
510 REM Initial positions - lower case for white
520 DIM b$(8,8,4)
530 LET b$(1,1)="abcd": LET b$(1,8)="abcd": REM rook
540 LET b$(1,2)="efgh": LET b$(1,7)="efgh": REM knight
550 LET b$(1,3)="ijkl": LET b$(1,6)="ijkl": REM bishop
560 LET b$(1,4)="mnoh": REM queen
570 LET b$(1,5)="pqcd": REM king
580 FOR n=1 TO 8: LET b$(2,n)="rstu": NEXT n: REM pawns
590 FOR n=1 TO 8: LET b$(7,n)="RSTU": NEXT n: REM black pieces
600 LET b$(8,5)="PQCD"
610 LET b$(8,4)="MNOH"
620 LET b$(6,3)="IJKL": LET b$(8,6)="IJKL"
630 LET b$(8,2)="EFGH": LET b$(8,7)="EFGH"
640 LET b$(8,1)="ABCD": LET b$(8,8)="ABCD"
650 IF NOT w THEN GO TO 1520: REM stps board printing
1000 REM
1010 REM print blank board
1020 BORDER 7: PAPER 7: INK 0: CLS
1030 PRINT TAB 8;"A B C D E F G H"
1040 FOR n=1 TO 8: PRINT AT 2*n,6;n: NEXT n
1100 FOR n=0 TO 3: FOR m=0 TO 3: FOR b=0 TO 1
1110 PRINT AT 2+4*n+b,6+4*m;"██": REM 2 magenta paper spaces followed by two green paper spaces
1120 PRINT AT 4+4*n+b,6+4*m;"██": REM coloured spaces the other way round
1130 NEXT b: NEXT m: NEXT n
1500 REM
1510 REM print pieces
1520 FOR v=1 TO 8: FOR n=1 TO 8: LET t=0
1530 BRIGHT 0: PAPER 0: INK 1: FOR m=1 TO 4
1540 LET b=CODE b$(v,n,m)
1550 IF b=CODE " " THEN GO TO 1580: REM one space
1560 IF b>90 THEN LET b=b-32: INK 7: REM lower case is white
1570 LET b=b+79: REM convert to graphics
1580 LET r=v*2+INT ((m-1)/2)
1590 LET s=6+2*n: IF INT (m/2)=m/2 THEN LET s=s+1
1600 PRINT AT r,s;CHR$ b
1610 NEXT m: IF t THEN RETURN
1615 IF w AND v=2 AND n=8 THEN LET v=6: REM stops space printing
1620 NEXT n: NEXT v
2000 REM
2010 REM enter moves
2020 DIM a$(4): INPUT "Enter your move as eg f1g3""If you wish to stop enter 0 ";a$
2030 IF a$(1)="0" THEN GO TO 2180: REM another game?
2040 LET n=CODE a$(1)
2050 IF n<73 THEN LET n=n-64: REM accepts capitals
2060 IF n>96 THEN LET n=n-96: REM and lower case
2070 LET v=VAL a$(2)
2080 LET n1=CODE a$(3)
2090 IF n1<73 THEN LET n1=n1-64
2100 IF n1>96 THEN LET n1=n1-96
2110 LET v1=VAL a$(4)
2120 IF v>8 OR v<1 OR n>8 OR n<1 OR v1>8 OR v1<1 OR n1>8 OR n1<1 THEN INPUT "Invalid move. Try again ";a$: GO TO 2030
2130 LET b$(v1,n1)=b$(v,n)
2140 LET b$(v,n)="    ": REM four spaces
2150 LET t=1: GO SUB 1530: REM erase piece
2155 IF b$(v1,n1,1)="r" AND v1=8 THEN LET b$(v1,n1)="mnoh": REM change pawn to queen
2156 IF b$(v1,n1,1)="R" AND v1=1 THEN LET b$(v1,n1)="MNOH"
2160 LET v=v1: LET n=n1: GO SUB 1530: REM print piece in new position
2170 GO TO 2020: REM next move
2180 INPUT "Do you want another game? y/n ";c$
2190 IF c$="y" OR c$="Y" THEN LET w=0: GO TO 500: REM reset board
2200 BRIGHT 0: INK 0: STOP
```

lower-case allows you to distinguish between black and white pieces. The dimension statement (520) resets the array automatically.

Lines 1000-1130 print the blank board using magenta and green. They are chosen to give maximum contrast and therefore clarity for the pieces, which are black and white for simplicity. It is spaces which are printed using colour control characters (see page 115 of the manual) to define their paper colour. Remember to switch back to white paper after the fourth space.

Lines 1500-1620 print the pieces on the board using black or white ink. Paper 8 keeps the board squares to their original colour. Lines 1580 and 1590 keep the print position within the correct 2*2 square.

Lines 2000-2200 enter the piece moves, checking at 2120 that the move is within the board and allowing you to stop the game. Either capitals or lower-case are accepted in inputs; 2140 unprints the old position and 2160 prints the new; 2155 and 2156 convert pawns to queens.

The variable set in 2150 allows use of earlier lines as a subroutine. It is re-set in 1520. Variable w (set in 260) allows the omission of some printing — blank squares on the initial run (line 1615) and reprinting the board (line 650) in subsequent games. w is re-sent at 2190.

THE THOUGHT PROCESSOR

MiCROL©
SPECTRUM
USE AND LEARN VOL.1: 25 BASIC PROGRAMS

AVAILABLE NOW

£9.95

USE 25 PRACTICAL BASIC programs you can put to work immediately–

Programs to demonstrate the wide-ranging potential of your 16 or 48K Spectrum:—

World Atlas — Cassette and Videocassette Index — Music Composer — Computer Term Glossary — Star Maps . . .

Personal Programming Aids to help you write your own programs more effectively:—

Memory Map Monitor — System Diagnostic — Program debugging aids . . .

Time-saving routines to use in your own programs:–

Text Editor — Flexible graph drawing routines — Sort and Search routines . . .

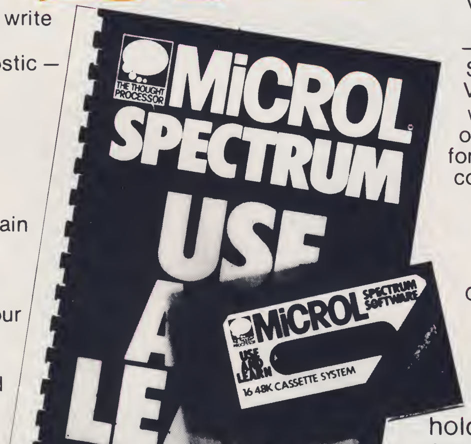
Plus much more.

And, of course, original games to entertain and challenge you.

LEARN New ways to get the most from your Spectrum. Over 100 pages packed with —

Powerful programming techniques:— use Structured Programming to save time and make your programs more reliable — Ideas to make your games more exciting — How and when to use trees, tables, sorts and searches (do you know the Monkey Puzzle sort?) . . .

Facts at your Fingertips:— Memory and runtime Benchmarks for every command — Display File Memory Map — Important PEEK and POKE locations you won't find in your Spectrum manual. Program Design Aids:–Screen Designers, for fast graphics and print layouts — Memory Manager, to keep track of every variable and array. PLUS:– All 25 BASIC programs explained line by line –a goldmine of practical hints and tips. Send today for USE AND LEARN Volume 1 – 25 BASIC Programs, and we'll also keep you posted with details of further important MiCROL products for your Spectrum. And USE AND LEARN comes with MiCROL's full 14-day money- back Guarantee.



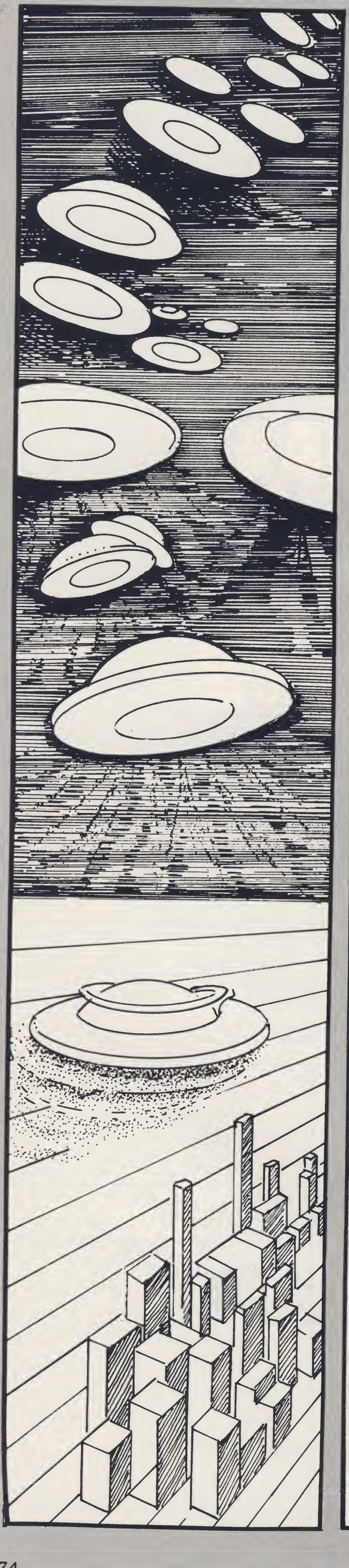
To order simply complete the coupon, and FREEPOST with your cheque, made payable to **MiCROL** (UK Mail Order).

Despatch normally by return.

Telephone orders—credit card holders can order by telephoning **(0223) 312866** from 9-5.30 Monday to Saturday, stating name and address, Card No. Access/Barclaycard/Visa. and item(s) required.

PROGRAM PRINTOUT

One of the most popular sections of *Sinclair User* has been Program Printout, in which we have attempted to give a wide variety of programs to suit all abilities, interests and machines. We reprint some of the best with some new programs never published previously.



SAUC

THE GRAPHICS on Saucers are the best we have seen for the old Earth Defence routine. Your task, as usual, is to halt the descending hordes of tentacled terror.

In this listing you are equipped with a movable gun-sight, under the usual cursor control, and if too many of these appalling aliens are allowed to land they erect their own

```
   3 GO SUB 5000
   4 GO SUB 4000
   5 LET l=15: LET k=11: LET pos
=10
   6 LET m=3: LET n=INT (RND*26)
+2
   8 LET score=0
   9 LET LAN=0
  10 PLOT 0,0: DRAW 97,50
  20 PLOT 255,0: DRAW -97,50
  30 PLOT 137,50: DRAW 50,-50
  40 PLOT 117,50: DRAW -50,-50
  50 PLOT 70,50: DRAW -70,-15
  60 PLOT 185,50: DRAW 70,-15
  70 PLOT 0,50: DRAW 255,0
  80 PLOT 127,50: DRAW 0,-50
  90 PLOT 0,40: DRAW 255,0
 100 PLOT 0,15: DRAW 255,0
 110 PLOT 0,14: DRAW 255,0
 120 INK 7: PLOT 0,0: DRAW 0,175
: DRAW 255,0: DRAW 0,-175: DRAW
-255,0
 125 PRINT AT k,l-1;" "
 126 PRINT AT k,l+1;" "
 127 PRINT AT k-1,l;" "
 128 PRINT AT k+1,l;" "
 130 IF INKEY$="8" AND l<29 THEN
 LET l=l+1
 140 IF INKEY$="5" AND l>2 THEN
LET l=l-1
 150 IF INKEY$="6" AND k<13 THEN
 LET k=k+1
 160 IF INKEY$="7" AND k>2 THEN
LET k=k-1
 180 PRINT AT INT m,INT n;" "
 200 PRINT AT k,l-1;"_"
 210 PRINT AT k-1,l;"|"
 220 PRINT AT k+1,l;"|"
 230 PRINT AT k,l+1;"_"
 240 IF INKEY$="0" THEN FOR t=0
TO 3: BEEP 0.02,40: NEXT t: IF k
=INT m AND l=INT n THEN GO SUB 1
000
 245 LET n=n+INT (RND*3)-1
 246 IF n<3 THEN LET n=n+1
 247 IF n>29 THEN LET n=n-1
 248 BEEP 0.005,(RND*35)+15
 250 LET m=m+0.2
 252 IF score>1000 THEN LET m=m+
0.2: IF m>2000 THEN LET m=m+0.2
 255 IF n<1 OR n>31 THEN PLOT 0,
0: DRAW 0,175: DRAW 255,0: DRAW
0,-175: DRAW -255,0
 260 IF m>15 THEN GO SUB 2000
 300 INK INT (RND*7)+2: PRINT AT
 INT m,INT n;"♣": INK 7
 310 PRINT AT 1,1;"SCORE:";score
;" "
 999 GO TO 125
1020 PRINT AT k,l;"♣": PRINT ; O
VER 1;AT k,l;"="
1030 FOR h=50 TO 45 STEP -0.5: B
EEP 0.01,h: NEXT h
1040 PRINT AT k,l;" "
1050 LET m=3: LET n=INT (RND*26)
+2
1060 LET score=score+100
1080 RETURN
2000 LET lan=lan+1
2010 IF lan>10 THEN GO TO 7000
```

CERS

own mucous-covered metropolis and use earthlings for toothpicks.

The most effective part of the display is the use of a perspective grid to represent the Earth's surface. An excellent program, to be run, recorded, revised. Submitted by N J Kemp of Liskeard, Cornwall, for the 16K Spectrum.

```
2015 LET m=3: LET n=INT (RND*25)
+2
2020 RETURN
3005 FOR n=0 TO 100: BEEP 0.1,40
: NEXT n
4000 PAPER 1: INK 7: BORDER 1: C
LS
4010: PRINT AT 0,10; FLASH 1;"SA
UCER"; FLASH 0
4020 INK 6: PRINT : PRINT "Save
the Earth.The aliens are   invad
ing and are trying to buildcitys
 on the Earth.": PRINT : PRINT "
Shoot them before they build
thier cities and conquer EARTH
and use you for tooth picks."
4030 INK 7: PRINT : PRINT : PRIN
T "  5 FOR LEFT": PRINT : PRINT
"  8 FOR RIGHT": PRINT : PRINT "
  6 FOR DOWN": PRINT : PRINT "
7 FOR UP": PRINT : PRINT "  0 TO
 FIRE"
4040 PRINT AT 21,1; FLASH 1;"Goo
d luck........": FLASH 0
4045 PAUSE 9999
4050 BORDER 0: PAPER 0: INK 6: C
LS : FOR n=0 TO 50: BEEP 0.005,(
RND*35)+15: INK INT (RND*7)+1: P
LOT INT (RND*250)+5,INT (RND*120
)+50: NEXT n: INK 4: GO TO 5
5000 DATA BIN 0,BIN 00011000,BIN
 01111110,BIN 11011011,BIN 11111
111,BIN 01100110,BIN 0,BIN 0
5010 FOR N=0 TO 7: READ J
5030 POKE USR "p"+n,J: NEXT n
5040 RETURN
7000 BEEP 0.7,1: BEEP 0.5,1: BEE
P 1,6
7010 PRINT AT 1,1;"  THEY HAVE C
ONQUERED EARTH"
7020 PRINT AT 2,3;"AND ARE BUILD
ING THEIR CITY"
7025 PAUSE 100
7030 INK INT 2: PRINT AT 12,10;"
█"
7035 BEEP 0.5,5
7040 PRINT AT 13,9;"▄█▄"
7045 BEEP 0.5,4
7050 PRINT AT 14,9;"███"
7055 BEEP 0.5,3
7060 PRINT AT 15,9;"███"
7065 BEEP 0.5,2
7070 PRINT AT 16,8;"▄███▄"
7075 BEEP 0.5,1
7076 INK 6
7080 PRINT AT 8,1;"YOU HAVE LOST
 EARTH. THE    "
7090 PRINT AT 9,1;"ALIENS HAVE D
OOMED MANKIND"
7100 PRINT AT 10,1;"TO BECOME TO
OTH PICKS"
7110 PRINT AT 18,1;"ENTER 'RUN'
TO GO BACK INTO  "
7120 PRINT AT 19,1;"TIME TO TRY
AND SAVE THE"
7130 PRINT AT 20,1;"EARTH AGAIN
!!!!"
7140 FLASH 1: PRINT AT 5,1;"YOUR
 SCORE WAS ";score: FLASH 0
```

FROM Daniel Shavick of Mill Hill, London, **Sheepdog** — a difficult game which is a good representation of a sheepdog trial.

A sheep, an inverse S, is driven by a dog, an inverse D, through a gate, denoted by two black squares, and into a pen which is shown as a grey square.The dog is moved upwards by pressing the 'O' key, downwards by the '.' key, left by 1 and right by 3. It continues moving until the S is pressed.

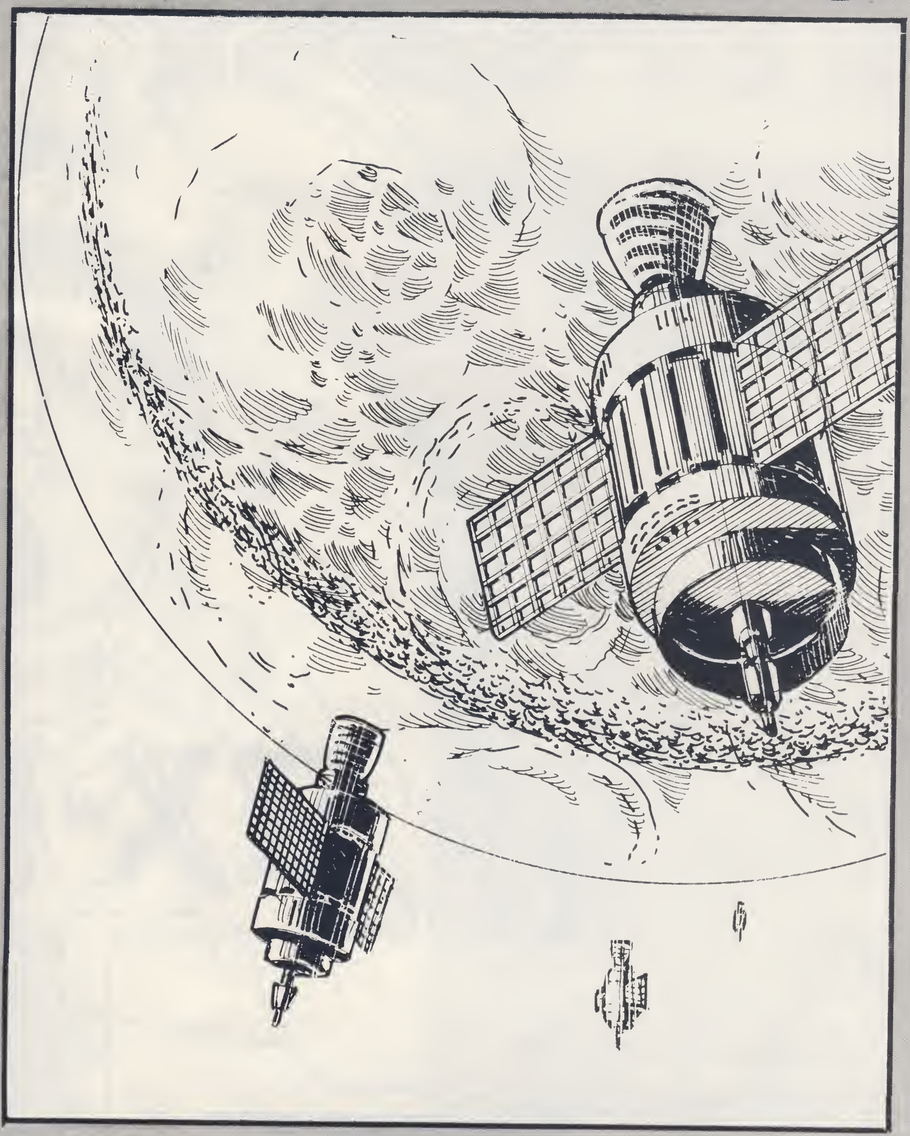
When the dog is within five squares of the sheep, the sheep begins to move. The difficulty is that the movement of the sheep tends to be as wayward as any sheep in a real trial.

After a good deal of concentration, the sheep can be penned and the time taken is displayed. As a guide it took our reviewer 1,079 seconds — one second short of 18 minutes.

The game can be re-started by pressing NEW LINE.

```
  10 PRINT AT 19,29;CHR$ 136;AT 10,5;CHR$ 128;AT 10,11;CHR$ 128
  20 LET S=0
  30 LET A$=""
  40 LET DH=21
  50 LET DL=5
  60 LET SH=5
  70 LET SL=25
  80 LET Z=5
  90 PRINT AT SH,SL;CHR$ 184;AT DH,DL;CHR$ 169
 100 IF SH=19 AND SL=29 THEN GOTO 1000
 110 LET S=S+1
 120 IF INKEY$<>"" THEN LET A$=INKEY$
 130 IF A$="" THEN GOTO 90
 140 PRINT AT DH,DL;CHR$ 0;AT SH,SL;CHR$ 0
 150 PRINT AT 19,29;CHR$ 136
 160 LET DH=DH+(A$=CHR$ 27 AND DH<21)-(A$=CHR$ 52 AND DH>0)
 170 LET DL=DL+(A$=CHR$ 31 AND DL<31)-(A$=CHR$ 29 AND DL>0)
 180 IF ABS (SH-DH)>=Z OR ABS (SL-DL)>=Z THEN GOTO 90
 190 IF ABS (DL-SL)<Z THEN LET SL=SL+(SL>DL)-(SL<DL)
 200 IF ABS (DH-SH)<Z THEN LET SH=SH+(SH>DH)-(SH<DH)
 210 LET SH=SH+(SH<1)-(SH>20)
 220 LET SL=SL+(SL<1)-(SL>30)
 230 IF SH=10 AND (SL<6 OR SL>10) THEN LET SH=SH-1
 240 GOTO 90
1000 PRINT AT 21,0;S;" SECONDS"
1010 PAUSE 4E4
1020 CLS
1030 RUN
```

EARTH DEFENCE



EARTH DEFENCE is an excellent arcade-qualtiy game for the 16K Spectrum. Adrian Tucker of Fareham, Hampshire, says he designed it to be equally comfortable for both left- and right-handed operators. He has done so by using the multi-key INKEY$ via the IN command in lines 110 and 160.

The effect is that your spacecraft can be manoeuvred left with any of the keys from 1 to 5, right with anything from 6 to 0, and all the bottom row keys will operate your laser.

A series of alien missiles moves up the screen towards earth. You are stationed in orbit to destroy the attackers. Each time you fire, your laser points are reduced by one; if you hit a missile, you have time to recharge and your laser points are therefore incremented by one.

You receive 100 points for each missile destroyed but lose 50 if one passes you. In the bottom left of the screen is displayed the number of alien projectiles yet to hit the earth before the limit of five is reached and the game ends.

The capital letters A, B and C in inverted commas are user-defined graphics. Tucker recommends blue for line 95, yellow for 115, cyan for 125, green for 140, flashing cyan in 145, red then cyan in 320 and green in 505.

```
   1 RUN 6
   2 FOR F=0 TO 7: READ N,M,O: P
OKE USR "B"+F,N: POKE USR "A"+F,
M: POKE USR "C"+F,O
   3 NEXT F
   4 DATA 16,165,165,16,189,24,5
6,255,165,124,255,90,186,126,90,
254,36,165,254,60,24,170,24,165
   5 STOP
  10 PAPER 0: INK 7: BORDER 1: C
LS
  20 LET HP=0
  50 CLS : FOR F=1 TO 50: PLOT I
NK RND*3+4,255.5*RND,175.5*RND:
NEXT f
  60 LET LA=50: LET L=1: PRINT A
T 21,0;"ALIENS";TAB 12;"PLAYER";
TAB 24;"LASER"
  70 LET P=0: LET C=16: LET CO=5
  80 GO SUB 504: LET L=0
  92 LET I=20: LET ST=INT (RND*2
0)+5
  93 IF CO=0 THEN GO TO 210
  95 PRINT AT I,ST;"B"
 100 PRINT AT 1,C;" "
 110 LET C=C+(IN 61438<>255 AND
C<31)-(IN 63486<>255 AND C>0)
 115 PRINT AT 1,C;"A"
 122 BEEP .005,C
 125 PRINT OVER 1;AT I,ST;"B"
 131 LET I=I-1
 133 IF I=-1 THEN GO TO 500
 136 LET ST=ST+INT (RND*3)-1+(3
AND ST<-30)-(3 AND ST>30)
 140 PRINT OVER 1;AT I,ST;"B"
 143 PRINT AT 21,19;P;" "
 145 IF I=1 AND ST=C THEN PRINT
AT 1,C;"C": GO TO 200
 160 IF (IN 65278<>255 OR IN 327
66<>255) AND LA>0 THEN LET LA=LA
-1: PRINT AT 21,30;LA: GO SUB 30
0
 165 IF LA<10 THEN PRINT AT 21,3
" "
 170 GO TO 100
 200 FOR G=1 TO 30
 205 OUT 254,RND*255
 206 BEEP RND*.05,RND*24-12
 208 NEXT G
 210 IF P>HP THEN LET HP=P
 221 PRINT AT 21,0
 222 INPUT ;"YOUR SCORE=";(P)''"
HI'SCORE=";(HP): PAUSE 500: GO T
O 1
 224 GO TO 1
 300 PLOT INVERSE 1,C*8+3,159
 305 DRAW INK 5,0,-117
 308 BEEP .05,12
 309 PLOT INVERSE 1,C*8+3,159
 310 DRAW OVER 1,0,-117
 320 IF (I<17 AND I>1) AND ST=C
THEN LET P=P+100: LET LA=LA+1: P
RINT AT 21,30;LA: BEEP .1,2: PRI
NT OVER 1;AT I,C;"C": BEEP .2,3:
 PRINT OVER 1;AT I,C;"C": BEEP .
1,4: PRINT AT I,C;" ": GO TO 85
 340 RETURN
 500 LET P=P-50: LET CO=CO-1
 503 PRINT AT 21,19;P;" "
 504 FOR F=1 TO 5
 505 IF F<CO THEN PRINT AT 21,6+
F;"B"
 508 IF F>CO THEN PRINT AT 21,5+
F;" "
 510 NEXT F: IF L=1 THEN RETURN
 520 GO TO 90
```

ZX-MAN

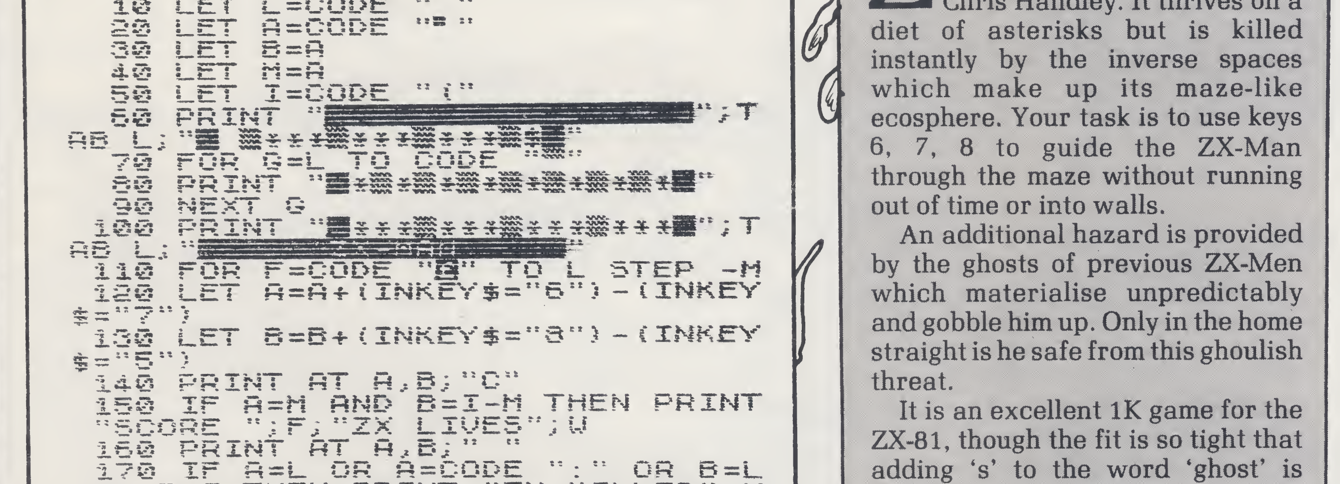
```
  10 LET L=CODE "."
  20 LET A=CODE "■"
  30 LET B=A
  40 LET M=A
  50 LET I=CODE "("
  60 PRINT "█████████████████";T
AB L;"█ ▒***▒***▒***▒*█"
  70 FOR G=L TO CODE "▒"
  80 PRINT "█*▒*▒*▒*▒*▒*▒*▒*█"
  90 NEXT G
 100 PRINT "█***▒***▒***▒***█";T
AB L;"██████ZX-MAN█████"
 110 FOR F=CODE "£" TO L STEP -M
 120 LET A=A+(INKEY$="6")-(INKEY
$="7")
 130 LET B=B+(INKEY$="8")-(INKEY
$="5")
 140 PRINT AT A,B;"C"
 150 IF A=M AND B=I-M THEN PRINT
"SCORE ";F;"ZX LIVES";U
 160 PRINT AT A,B;" "
 170 IF A=L OR A=CODE ":" OR B=L
OR B=I THEN PRINT "ZX KILLED";U
 180 LET E=INT (RND*CODE "£")
 190 IF E=A AND B<A THEN PRINT A
T A,B;"G";TAB L;"GHOST GOTCHA";U
 200 NEXT F
 210 PRINT "TIME UP"
```

ZX-MAN is a silicon-based life form, writes its discoverer, Chris Handley. It thrives on a diet of asterisks but is killed instantly by the inverse spaces which make up its maze-like ecosphere. Your task is to use keys 6, 7, 8 to guide the ZX-Man through the maze without running out of time or into walls.

An additional hazard is provided by the ghosts of previous ZX-Men which materialise unpredictably and gobble him up. Only in the home straight is he safe from this ghoulish threat.

It is an excellent 1K game for the ZX-81, though the fit is so tight that adding 's' to the word 'ghost' is sufficient to halt the program. Graphics notes:

60 — Seventeen inverse spaces.
100 — Six inverse spaces, inverse 'ZX-MAN', five, inverse spaces.
180 — Inverse £.
190 — Inverse G.



BEAN CUP

```
10 DIM A(14)
100 FOR J = 1 to 14
110 IF J = 7 OR J = 14 THEN GO
 TO 130
120 LET A (J) = 4
130 NEXT J
200 FOR J = 1 TO 7
210 PRINT A (15-J); "2 spaces";
220 NEXT J
230 PRINT
240 PRINT
250 PRINT "4 spaces";
260 FOR J = 1 TO 7
270 PRINT A (J); "2 spces";
280 NEXT J
290 PRINT
300 PRINT
310 PRINT "ENTER CUP NO."
320 INPUT C
330 CLS
340 GO SUB 1000
400 FOR J = 1 TO 6
410 IF NOT A (14-J) = 0 THEN
 GO TO 440
420 NEXT J
430 GO TO 460
440 LET C = 14-J
450 GO SUB 1000
460 IF A(7) + A(14)<48 THEN GO
 TO 200
470 IF A(7) = A(14) THEN PRINT
 "A DRAW"
480 IF A(7)>A(14) THEN PRINT
 "YOU WIN"
490 IF A(7)<(14) THEN PRINT "I
 WIN"
500 STOP
1000 IF A(C) = 0 THEN RETURN
1010 FOR J = C + 1 TO A(C) + C
1020 LET K = J
1030 IF J>14 THEN LET K = J-14
1040 LET A(K) = A(K) + 1
1050 NEXT J
1060 LET A(C) = 0
1070 IF NOT A(K) = 1 OR NOT
  K = 7 OR NOT K = 14 THEN
  RETURN
1080 LET A(K) = A(K) + A(14-K)
1090 LET A(14-K) = 0
2000 RETURN
```

SOMETIMES, when the cursor finger is worn to the knuckle and we are blinking at 50 cycles per second, nothing pleases us like a game where thinking is not just something between you and a high score. When the game works on a ZX-80, we are really happy.

Beancup is a fine brain-game from Nigeria. You sit cross-legged with a row of seven cups in front of you. Each cup contains four beans — except the store-cup on the right-hand end of the row, which is empty. Your opponent faces you with exactly the same equipment in front of him.

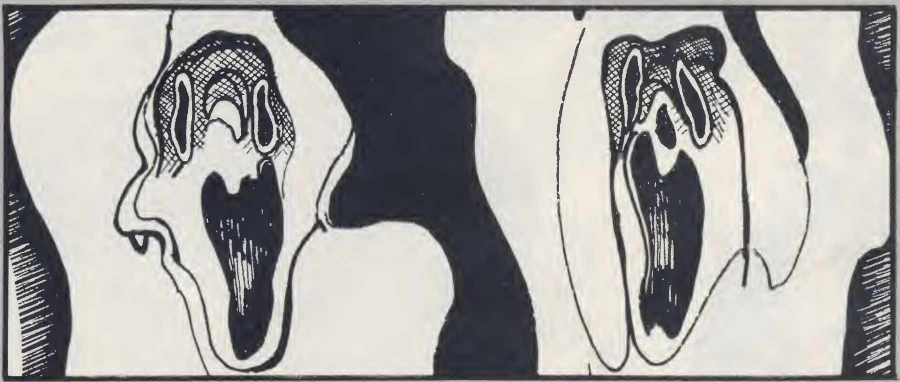
The display will show you two rows of numbers, representing the beans. The bottom row is yours and the top one is operated by the suddenly-cunning ZX-80.

You move by taking all the beans from one of your cups and dropping one into the cup on its right, one into the cup on the right of that, and so on, anti-clockwise round all 14 cups until you finish. You cannot empty the store-cups. The game ends when all the beans are out of circulation and the winner is the player with the most beans in his store cup.

Input the number of the cup you wish to empty and the computer will display the position of the beans after your move and its own response. If you have no beans on your side to move, then enter any number from 1 to 6.

If you can beat the machine you are better than we are. Submitted by Paul Morriss of Alford, Lincs.

SPACMAN



HERE is a good version of the perennial Pacman from S J Stearn of Harpenden, Herts for the 16K Spectrum.

Naturally it suffers by comparison with arcade machines. There are no power pills, no exit, and the two ghosts are decidedly more dumb than their cousins in the amusement halls, though still fast enough to catch you more times than not.

This listing is a big leap forward in the quest for an arcade-quality Pacman and provides a good basis for anyone wanting to do more work on the project; providing exits, for example, would probably not be too difficult.

```
   1 GO SUB 9000
   2 LET R=.01: LET s=0: LET gx=
10: LET gx1=2: LET gy1=2: LET gy
=10: LET x=19: LET y=2: LET d$="
◖": BORDER 1: PAPER 1: INK 1: CL
S : PAPER 7
   9 DIM a$(20,20)
  10 LET a$(1)="████████████████
████"
  20 LET a$(2)="█.....██████....
...█"
  30 LET a$(3)="█.████.█████.███
██.█"
  40 LET a$(4)="█..██.█.████...█
█..█"
  50 LET a$(5)="██....█..██..█..
..██"
  60 LET a$(6)="█.█.███.....████
█..█"
  70 LET a$(7)="█.████████.██...
..█"
  80 LET a$(8)="█......███...████
█.█"
  90 LET a$(9)="████████..█.█...█
█..█"
 100 LET a$(10)="█......███.███..
.██.█"
 110 LET a$(11)="████.████....██.
███..█"
 120 LET a$(12)="█....███.......
██.█"
 130 LET a$(13)="█.██████.....██.
....█"
 140 LET a$(14)="█.....██████.█..█.
█.███"
 150 LET a$(15)="███.██.....███...
....█"
 160 LET a$(16)="█..........███.██.
█..█"
 170 LET a$(17)="███.██████.███..
.██.█"
 180 LET a$(18)="███.███.........
...██"
 190 LET a$(19)="█.....███.....███
█..█"
 200 LET a$(20)="████████████████
████"
 210 PRINT ': FOR f=1 TO 20: PRI
NT "█";a$(f): NEXT f
1000 LET a$(x,y)=" "
1001 IF S=180 OR S=460 OR S=740
THEN LET S=S+100: PAPER 1: CLS :
 GO SUB 7000: PAPER 1: CLS : PAP
ER 7: GO TO 10
1010 IF INKEY$="" THEN GO TO 105
5
1020 IF INKEY$="5" THEN LET d$="
◗"
1030 IF INKEY$="6" THEN LET d$="
▲"
1040 IF INKEY$="7" THEN LET d$="
▼"
1050 IF INKEY$="8" THEN LET d$="
◖"
1055 PRINT AT x,y;" "
1060 IF d$="▼" AND a$(x-1,y)<>"█
" THEN LET x=x-1
1061 IF d$="◖" AND a$(x,y+1)<>"█
" THEN LET y=y+1
1062 IF d$="▲" AND a$(x+1,y)<>"█
" THEN LET x=x+1
1063 IF d$="◗" AND a$(x,y-1)<>"█
" THEN LET y=y-1
1064 PRINT AT X,Y; INK 0;D$
1070 IF a$(x,y)="." THEN LET s=s
+1: PRINT AT 0,0;s: BEEP .1,-10
1080 PRINT AT gx,gy;a$(gx,gy): I
F INT (RND*2)+(GX>X) AND a$(gx-1
,gy)<>"█" THEN LET gx=gx-1
1090 IF INT (RND*2)+(GX<X) AND a
$(gx+1,gy)<>"█" THEN LET gx=gx+1
2000 IF INT (RND*2)+(GY>Y) AND a
$(gx,gy-1)<>"█" THEN LET gy=gy-1
2010 IF INT (RND*2)+(GY<Y) AND a
$(gx,gy+1)<>"█" THEN LET gy=gy+1
2015 PRINT AT gx,gy; INK 3;"♠"
2020 IF (gx=x AND gy=y) OR (gx1=
x AND gy1=y) THEN PRINT AT x,y;
FLASH 1;d$: STOP
3080 PRINT AT gx1,gy1;a$(gx1,gy1
): IF INT (RND*3)+(GX1>X) AND a$
(gx1-1,gy1)<>"█" THEN LET gx1=gx
1-1
3090 IF INT (RND*2)+(GX1<X) AND
a$(gx1+1,gy1)<>"█" THEN LET gx1=
gx1+1
3100 IF INT (RND*2)+(GY1>Y) AND
a$(gx1,gy1-1)<>"█" THEN LET gy1=
gy1-1
3110 IF INT (RND*2)+(GY1<Y) AND
a$(gx1,gy1+1)<>"█" THEN LET gy1=
gy1+1
3120 PRINT AT gx1,gy1; INK 4;"♠"
3130 GO TO 1000
7000 FOR F=0 TO 25
7001 PRINT AT 10,29; INK 7;"0"
7010 PRINT AT 10,F; INK 3;" ♠  "
; INK 6;"◖"
7020 BEEP .1,-15
7030 NEXT F
7035 BEEP .2,-20
7040 FOR F=25 TO 0 STEP -1
7050 PRINT AT 10,F; INK 5;"♠  ";
 INK 6;"◗ "
7055 BEEP 0.1,0
7060 NEXT F: CLS : RETURN
8999 STOP
9000 DATA 0,BIN 1000010,BIN 1110
0111,255,255,BIN 1111110,BIN 111
100,BIN 11000
9010 DATA BIN 00011100,BIN 1111
10,BIN 1111100,BIN 1111000,BIN 1
111000,BIN 1111100,BIN 111110,BI
N 11100
9020 DATA 0,BIN 00011000,BIN 001
11100,BIN 01111110,255,255,BIN 1
1100111,BIN 01000010,0
9030 DATA BIN 001110000,BIN 0111
11000,BIN 001111100,BIN 0001111
0,BIN 00011110,BIN 00111100,BI
N 01111000,BIN 00110000
9035 DATA 24,60,126,219,219,255,
219,145
9040 RESTORE 9000
9045 FOR f=0 TO 7: READ a: POKE
USR "u"+f,a: NEXT f
9050 FOR f=0 TO 7: READ a: POKE
USR "r"+f,a: NEXT f
9060 FOR f=0 TO 7: READ a: POKE
USR "d"+f,a: NEXT f
9070 FOR f=0 TO 7: READ a: POKE
USR "l"+f,a: NEXT f
9075 FOR f=0 TO 7: READ a: POKE
USR "a"+f,a: NEXT f
9080 RETURN
```

TRIGONOMETRY

```
  10 CLS
  20 PRINT AT 1,6;"**TRIGONOMETR
Y**"
  30 PRINT AT 4,0;"INPUT";TAB 8;
"FORMULAE"
  40 PRINT AT 6,2;"A";TAB 7;"SIN
=OPP/HYP";TAB 27;"■"
  50 PRINT AT 7,2;"B";TAB 7;"OPP
=SIN*HYP";TAB 22;"HYP ■ OPP"
  60 PRINT AT 8,2;"C";TAB 7;"HYP
=OPP/SIN  SIN->■"
  70 PRINT AT 10,2;"D";TAB 7;"CO
S=ADJ/HYP";TAB 27;"■"
  80 PRINT AT 11,2;"E";TAB 7;"AD
J=COS*HYP";TAB 22;"HYP ■"
  90 PRINT AT 12,2;"F";TAB 7;"HY
P=ADJ/COS";TAB 20;"COS->■"
 100 PRINT TAB 25;"ADJ"
 110 PRINT AT 15,2;"G";TAB 7;"TA
N=OPP/ADJ";TAB 27;"■"
 120 PRINT AT 16,2;"H";TAB 7;"OP
P=TAN*ADJ";TAB 26;"■ OPP"
 130 PRINT AT 17,2;"I";TAB 7;"AD
J=OPP/TAN";TAB 20;"TAN->■"
 140 PRINT TAB 25;"ADJ";AT 19,2;
"J";TAB 7;"■"
 150 PRINT AT 21,0;"INPUT: REQUI
RED FORMULAE(A TO J)"
 800 INPUT B$
 810 IF B$="A" THEN GOTO 1100
 820 IF B$="B" THEN GOTO 1300
 830 IF B$="C" THEN GOTO 1500
 840 IF B$="D" THEN GOTO 1700
 850 IF B$="E" THEN GOTO 1900
 860 IF B$="F" THEN GOTO 2100
 870 IF B$="G" THEN GOTO 2300
 880 IF B$="H" THEN GOTO 2500
 890 IF B$="I" THEN GOTO 2700
 900 IF B$="J" THEN STOP
 910 IF B$<"A" OR B$>"J" THEN GO
TO 800
1100 CLS
1110 PRINT AT 2,2;"INPUT OPP"
1120 INPUT O
1130 PRINT AT 2,15;"OPP= ";O
1140 PRINT AT 4,2;"INPUT HYP"
1150 INPUT H
1160 PRINT AT 4,15;"HYP= ";H
1170 LET S=O/H
1180 LET X=ASN S*180/PI
1190 PRINT AT 6,5;"ANGLE= ";X;"
DEGS"
1200 PRINT AT 10,3;"PRESS NEWLIN
E TO CONTINUE"
1210 INPUT A$
1220 GOTO 10
1300 CLS
1310 PRINT AT 2,2;"INPUT SIN"
1320 INPUT S
1330 PRINT AT 2,15;"SIN= ";S;" D
EGS"
1340 PRINT AT 4,2;"INPUT HYP"
1350 INPUT H
1360 PRINT AT 4,15;"HYP= ";H
1370 LET O=(SIN (S*PI/180))*H
1380 PRINT AT 6,5;"OPP= ";O
1390 PRINT AT 10,3;"PRESS NEWLIN
E TO CONTINUE"
1400 INPUT A$
1410 GOTO 10
1500 CLS
1510 PRINT AT 2,2;"INPUT OPP"
1520 INPUT O
1530 PRINT AT 2,15;"OPP= ";O
1540 PRINT AT 4,2;"INPUT SIN"
1550 INPUT S
1560 PRINT AT 4,15;"SIN= ";S;" D
EGS"
1570 LET H=O/(SIN (S*PI/180))
1580 PRINT AT 6,5;"HYP= ";H
```

WHICH TYPE of child were you? The type who instinctively hides at the back of the class, whispering and flicking paper? Or the kind who rushes automatically to the front and sits, tense and eager, taking copious notes with the sharpest pencil in school?

P L Radford of Birmingham sent a program which will be useful for both types. The first type will appreciate a program which reminds you how to do all those tricky trigonometry equations and the second will be glad of a way of doing them more quickly.

A thorough and useful listing for the 16K ZX-81, to be taped and kept.

```
1590 PRINT AT 10,3;"PRESS NEWLIN
E TO CONTINUE"
1600 INPUT A$
1610 GOTO 10
1700 CLS
1710 PRINT AT 2,2;"INPUT ADJ"
1720 INPUT A
1730 PRINT AT 2,15;"ADJ= ";A
1740 PRINT AT 4,2;"INPUT HYP"
1750 INPUT H
1760 PRINT AT 4,15;"HYP= ";H
1770 LET C=A/H
1780 LET X=ACS C*180/PI
1790 PRINT AT 6,5;"ANGLE= ";X;"
DEGS"
1800 PRINT AT 10,3;"PRESS NEWLIN
E TO CONTINUE"
1810 INPUT A$
1820 GOTO 10
1900 CLS
1910 PRINT AT 2,2;"INPUT COS"
1920 INPUT C
1930 PRINT AT 2,15;"COS= ";C;" D
EGS"
1940 PRINT AT 4,2;"INPUT HYP"
1950 INPUT H
1960 PRINT AT 4,15;"HYP= ";H
1970 LET A=(COS (C*PI/180))*H
1980 PRINT AT 6,5;"ADJ= ";A
1990 PRINT AT 10,3;"PRESS NEWLIN
E TO CONTINUE"
2000 INPUT A$
2010 GOTO 10
2100 CLS
2110 PRINT AT 2,2;"INPUT ADJ"
2120 INPUT A
2130 PRINT AT 2,15;"ADJ= ";A
2140 PRINT AT 4,2;"INPUT COS"
2150 INPUT C
2160 PRINT AT 4,15;"COS= ";C;" D
EGS"
2170 LET H=A/(COS (C*PI/180))
2180 PRINT AT 6,5;"HYP= ";H
2190 PRINT AT 10,3;"PRESS NEWLIN
E TO CONTINUE"
2200 INPUT A$
2210 GOTO 10
2300 CLS
2310 PRINT AT 2,2;"INPUT OPP"
2320 INPUT O
2330 PRINT AT 2,15;"OPP= ";O
2340 PRINT AT 4,2;"INPUT ADJ"
2350 INPUT A
2360 PRINT AT 4,15;"ADJ= ";A
2370 LET T=O/A
2380 LET X=ATN T*180/PI
2390 PRINT AT 6,5;"ANGLE= ";X;"
DEGS"
2400 PRINT AT 10,3;"PRESS NEWLIN
E TO CONTINUE"
2410 INPUT A$
2420 GOTO 10
2500 CLS
2510 PRINT AT 2,2;"INPUT TAN"
2520 INPUT T
2530 PRINT AT 2,15;"TAN= ";T;" D
EGS"
2540 PRINT AT 4,2;"INPUT ADJ"
2550 INPUT A
2560 PRINT AT 4,15;"ADJ= ";A
2570 LET O=(TAN (T*PI/180))*A
2580 PRINT AT 6,5;"OPP= ";O
2590 PRINT AT 10,3;"PRESS NEWLIN
E TO CONTINUE"
2600 INPUT A$
2610 GOTO 10
2700 CLS
2710 PRINT AT 2,2;"INPUT OPP"
2720 INPUT O
2730 PRINT AT 2,15;"OPP= ";O
2740 PRINT AT 4,2;"INPUT TAN"
2750 INPUT T
2760 PRINT AT 4,15;"TAN= ";T;" D
EGS"
2770 LET A=O/(TAN (T*PI/180))
2780 PRINT AT 6,5;"ADJ= ";A
2790 PRINT AT 10,3;"PRESS NEWLIN
E TO CONTINUE"
2800 INPUT A$
2810 GOTO 10
```

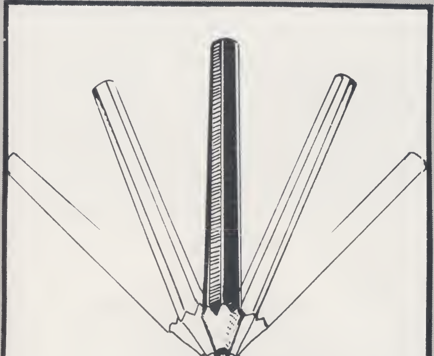
MAGIC SQUARES

```
 5 PRINT "INPUT THE NO."
 6 INPUT M
 7 SCROLL
12 FOR F=0 TO 11
13 PRINT AT 7,F;"■";AT 0,F;"■"
14 NEXT F
15 FOR F=0 TO 7
16 PRINT AT F,0;"■";AT F,12;"■

17 NEXT F
21 RAND
22 LET L=INT (RND*20)
25 PRINT AT 1,1;L
30 LET K=L+10
35 PRINT AT 3,1;K
40 LET J=M-(L+K)
45 PRINT AT 5,1;J
46 LET H=INT (RND*24)
50 PRINT AT 1,5;H
55 LET G=M-(L+H)
56 PRINT AT 1,8;G
60 LET F=INT (RND*40)
65 PRINT AT 3,5;F
66 LET D=M-(H+F)
67 PRINT AT 5,5;D
70 LET S=M-(K+F)
75 PRINT AT 3,8;S
81 LET A=M-(J+D)
85 PRINT AT 5,8;A
90 PRINT AT 8,0;"MAGIC SQUARE"
95 PRINT "OF=";M
```

AVIAD RAZ, of Haifa, Israel, sent Magic Squares, a neat, quick 1K program for the ZX-81. It calculates and displays magic squares, a box of nine numbers which add to the same figure whether added horizontally or vertically.

He explains that 1K of memory is not sufficient to allow the square to work diagonally.



```
   5 LET LX=0: LET LY=0
   7 LET X=0: LET Y=0
  10 CLS
  20 INPUT "RADIUS OF OUTER RING
?";R
  25 IF R=0 THEN GO TO 10
  30 INPUT "RADIUS OF WHEEL?";R1
  40 INPUT "POSITION OF WHEEL?";
D
  50 INPUT "SPEED?";S
  55 LET LX=127: LET LY=87+R-D
  57 PLOT LX,LY
  60 LET A=0
 100 LET X=(R-R1)*SIN (A)+(R1-D)
*SIN (A*-(R/R1-1))+127
 110 LET Y=(R-R1)*COS (A)+COS (A
*-(R/R1-1))*(R1-D)+87
 120 DRAW X-LX,Y-LY
 130 LET LX=X
 140 LET LY=Y
 150 LET A=A+S
 160 IF INKEY$="" THEN GO TO 100
 180 GO TO 20
```

SPIROGRAPH

SPIROGRAPH, from Martin Richards of Stevenage, Herts, uses the hi-res graphics of the Spectrum to fascinating effect. The program produces graphics in the style of those line-drawing machines with the swinging weights and turning tables. On the Spectrum, too, it is fun to watch and you have good pictures.

```
   5 LET SX=0
   6 LET SB=0
   7 LET O=PEEK 16396+256*PEEK 1
6397
  10 POKE 16418,0
  20 PRINT AT 0,0;"▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒"
  30 FOR P=1 TO 22
  40 PRINT "▒                              ▒"
  50 NEXT P
  60 PRINT ;"▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒"
  70 PRINT AT 2,2;STR$ SX;AT 2,2
8;STR$ SB
  80 LET X=12
  90 LET Y=3
 100 LET B=12
 110 LET C=28
 120 LET X1=0
 130 LET Y1=1
 140 LET B1=0
 145 LET C1=-1
 150 IF PEEK (O+1+Y+33*X)<>0 THE
N GOTO 400
 160 IF PEEK (O+1+C+33*B)<>0 THE
N GOTO 300
 170 PRINT AT X,Y;"▓";AT B,C;"▒"
 180 LET A=PEEK 16421
 185 IF A=238 OR A=222 OR A=190
OR A=254 OR A=126 THEN LET X1=1
 190 IF A=247 OR A=231 OR A=215
OR A=183 OR A=119 THEN LET X1=-1
 200 IF X1<>0 THEN LET Y1=0
 210 IF A=235 OR A=251 OR A=219
OR A=187 OR A=123 THEN LET Y1=1
 215 IF A=237 OR A=221 OR A=253
OR A=189 OR A=125 THEN LET Y1=-1
 220 IF Y1<>0 THEN LET X1=0
 230 IF A=231 OR A=239 OR A=235
OR A=237 OR A=238 THEN LET B1=-1
 235 IF A=119 OR A=123 OR A=125
OR A=126 OR A=127 THEN LET B1=1
 240 IF B1<>0 THEN LET C1=0
 245 IF A=215 OR A=219 OR A=223
OR A=221 OR A=222 THEN LET C1=1
 250 IF A=183 OR A=187 OR A=189
OR A=191 OR A=190 THEN LET C1=-1
 260 IF C1<>0 THEN LET B1=0
 270 LET X=X+X1
 280 LET Y=Y+Y1
 290 LET B=B+B1
 293 LET C=C+C1
 295 GOTO 150
 300 LET SX=SX+1
 310 IF SX<15 THEN GOTO 7
 320 STOP
 400 LET SB=SB+1
 410 IF SB<15 THEN GOTO 7
```

SURROUND

YOU will probably recognise Surround as soon as you RUN it. It is a version of a fairly standard routine in which you are required to surround your adversary before he can surround you. Both players are in continuous motion and are not allowed to cross their tracks or those of their opponent, or to hit the boundary or the score display.

What will surprise you is the way the ZX-81 can recognise two keys when pressed simultaneously. That is not possible using INKEY$ and is achieved by utilising address 16421; which contains a number which changes as groups of keys are pressed. Lines 185 to 260 translate the number into a series of possible moves.

Black uses keys 1-5 to move up, Z-V to move down, Q-T for right and A-G for left. Grey uses the keys 6-0 for up, Y-P for right, B-M for down and H-NEWLINE for left. After a crash the other player scores 1 and the winner is the first to reach 15.

The routine enabling two players to participate obviously can be used in many graphics games.

TOWERS of HANOI

THIS game is from the mysterious east via Simon Annetts of Rhayader, Powys. It is for the 16K ZX-81 and is velly, velly good.

Your Sinclair will erect three pegs and on the centre peg will place six rings in order of descending size. Your task is to transfer the rings one by one until they are all assembled in the correct order on one of the other pegs.

You cannot place a bigger ring on a smaller one and the computer will chastise you if you try. An excellent little game which went immediately on to the office tape. Graphics notes:

30 — 32 graphic shifted 8s.
50 — Three spaces, graphic shifted 8,7 spaces, graphic shifted 8, seven spaces, graphic shifted 8.
67 — Three spaces, graphic shifted 8.
70 — Three spaces, inverse space, graphic shifted 5.
80 — Two spaces, graphic shifted 8, two inverse spaces.
90 — Two spaces, three inverse spaces, graphic shifted 5.
100 — One space, graphic shifted 8, four inverse spaces.
110 — One space, five inverse spaces, graphic shifted 5.
120 — Graphic shifted 8, six inverse spaces.

```
  20 CLS
  30 PRINT AT 20,0;"▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒"
  35 PRINT AT 21,8;"1";TAB 16;"2
";TAB 24;"3"
  40 FOR X=10 TO 19
  50 PRINT AT X,5;"   ▒       ▒       ▒"
  60 NEXT X
  61 DIM A(3,7)
  65 DIM A$(7,8)
  67 LET A$(1)="   ▒"
  70 LET A$(2)="   █▌"
  80 LET A$(3)="  ▒██"
  90 LET A$(4)="  ███▌"
 100 LET A$(5)=" ▒████"
 110 LET A$(6)=" █████▌"
 120 LET A$(7)="▒██████"
 130 FOR Z=7 TO 1 STEP -1
 135 LET A(1,Z)=1
 140 LET A(2,Z)=Z
 145 LET A(3,Z)=1
 150 NEXT Z
 155 LET C=1
 160 FOR Z=1 TO 3
 170 FOR Y=7 TO 1 STEP -1
 180 PRINT AT Y+12,Z*8-3;A$(A(Z,
Y))
 190 NEXT Y
 195 NEXT Z
 201 PRINT AT 0,24;"MOVE:";C
 205 IF A(1,2)=2 OR A(3,2)=2 THE
N GOTO 2000
 208 PRINT AT 0,0;"                    "
 210 PRINT AT 0,0;"FROM? "
 220 INPUT J
 230 PRINT AT 0,0;J;" TO ?"
 240 INPUT K
 250 PRINT AT 0,0;J;" TO ";K
 255 IF J>3 OR J<1 OR K>3 OR K<1
THEN GOTO 1000
 260 IF K=J THEN GOTO 1000
 270 FOR D=1 TO 7
 280 IF A(J,D)=1 THEN GOTO 320
 290 LET P=D
 300 LET Q=A(J,D)
 310 GOTO 340
 320 NEXT D
 330 GOTO 1000
 340 FOR D=1 TO 7
 350 IF A(K,D)=1 THEN GOTO 370
 360 IF A(K,D)<Q THEN GOTO 1000
 365 IF A(K,D)>1 THEN GOTO 380
 370 NEXT D
 380 LET D=D-1
 390 LET A(K,D)=A(J,P)
 400 LET A(J,P)=1
 410 LET C=C+1
 420 GOTO 160
1000 FOR U=0 TO 50
1020 PRINT AT 0,0;"INVALID MOVE"
1040 NEXT U
1050 GOTO 208
2000 PRINT AT 0,0;"YOU HAVE COMP
LETED THE TOWERS OF HANOI IN ";C
-1;" MOVES"
2010 PRINT
2020 PRINT "TRY AGAIN?"
2030 INPUT S$
2040 IF S$(1)="Y" THEN RUN
2050 STOP
```

THIS PROGRAM simulates the random decay of atoms. When run, it displays 256 atoms and a bar chart; as one atom decays, a new one is born. The bar chart keeps a record of the number of mother and daughter atoms. Every decay is also marked by a burst of sound. The rate should slow if the decay is random.

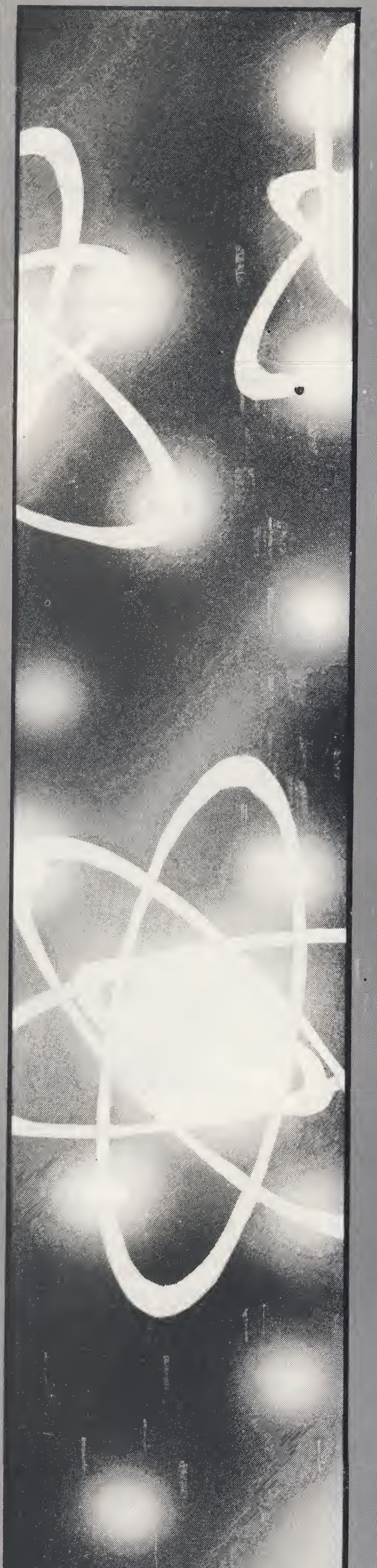
Following the decay of the final atom you are asked to guess the half-life of the atom. Your standing as a nuclear scientist is then analysed and you are told how close you were to the answer.

Radioactive Decay was sent by D Noonan, of London N8.

RADIO ACTIVE DECAY

```
  10 CLEAR 32499
  20 BORDER 7: PAPER 7: CLS
  30 PRINT AT 0,1; PAPER 5; INK
0;"RADIOACTIVE DECAY SIMULATION"
  40 LET y=3
  50 FOR x=1 TO 18: PRINT AT y,x
; INK 3;"█": NEXT x
  60 LET y=20
  70 FOR x=1 TO 18: PRINT AT y,x
; INK 3;"█": NEXT x
  80 LET x=1
  90 FOR y=4 TO 19: PRINT AT y,x
; INK 3;"█": NEXT y
 100 LET x=18
 110 FOR y=4 TO 19: PRINT AT y,x
; INK 3;"█": NEXT y
 120 FOR x=2 TO 17
 130 FOR y=4 TO 19
 140 PRINT AT y,x; INK 7; PAPER
1;"#"
 150 NEXT y
 160 NEXT x
 170 DIM m(100)
 180 LET a=0
 190 INK 0: PLOT 175,144: DRAW 0
,-129: DRAW 64,0
 200 FOR y=4 TO 19: PRINT AT y,2
3; INK 1;"█": NEXT y
 210 PRINT AT 2,1; INK 2;"TIME";
AT 3,21; INK 1;"#";AT 3,25; INK
2;"!";AT 21,0; PAPER 6; INK 0;"P
ress G to see a graph of decay"
 220 LET mother=256
 230 LET daughter=0
 240 LET time=0
 250 LET r=INT (4*RND)
 260 LET x=2+(r+15)*RND
 270 LET y=4+15*RND
 280 IF ATTR (y,x)=15 THEN LET m
other=mother-1: LET daughter=dau
ghter+1: BEEP 0.1,50*RND: PRINT
AT y,x; PAPER 4+r; INK 2; BRIGHT
 1;"!": GO TO 300
 290 PAUSE 4: PRINT AT y,x; OVER
 1; BRIGHT 1; PAPER 8; INK 8;" "
 300 PAPER 7: INK 1: PLOT INVERS
E 1;184,mother/2+15: DRAW INVERS
E 1;7,0
 310 INK 2: PLOT 208,daughter/2+
15: DRAW 8,0
 320 PRINT AT 3,22;"    "
 330 PRINT AT 3,22; INK 1;mother
;AT 3,26; INK 2;daughter
 340 LET time=time+1
 350 PRINT AT 2,6; INK 2;INT (ti
me/3)
 370 IF time/12=INT (time/12) TH
EN GO SUB 410
 380 IF PEEK 23560=103 THEN PRIN
T AT 21,0;" Please wait it's com
ing soon  ": GO TO 580
 390 IF mother=0 THEN GO TO 450
 400 GO TO 260
 410 LET a=a+1
 420 IF a>100 THEN RETURN
 430 LET m(a)=INT (mother/2+.5)
 440 RETURN
 450 RESTORE 460: PRINT AT 21,0;
"
 "
 460 DATA 33,0,88,17,0,3,14,150,
113,35,27,122,179,32,249,201
 470 FOR f=1 TO 16
 480 READ a: POKE 32500+f,a: NEX
T f
 490 RANDOMIZE USR 32501
 500 BORDER 2
 510 PAUSE 300
 520 DATA 1,9,1,10,1,14,1,12,0.5
,10,0.5,9,1,7,1,8,1,5,1,7,1,9,.5
,10,.5,9,2,7
 530 RESTORE 520
 540 FOR f=1 TO 14
 550 READ d,n
 560 BORDER INT (n/2)
 570 BEEP .25*d,n: NEXT f
 580 FOR g=1 TO 1000
 590 NEXT g
 600 BORDER 7
 610 DATA 0,0,0,0,128,128,128,12
8
 620 RESTORE 610
 640 FOR f=0 TO 7
 650 READ u: POKE USR "a"+f,u
```

```
 660 NEXT f
 670 CLS : RESTORE 690
 680 INK 1: PRINT AT 0,5;"Graph
to show decay"
 690 DATA " 0"," 40"," 80","120"
,"160","200","240","280","N","o"
,".","o","f"," ","a","t","o","m"
,"s"
 700 FOR y=17 TO 3 STEP -2
 710 READ g$: PRINT AT y,1;g$;"-
": NEXT y
 720 FOR y=4 TO 14: READ a$: PRI
NT AT y,0;a$: NEXT y
 730 FOR x=5 TO 30 STEP 5
 740 PRINT AT 17,x;"' ";AT 18,x-1
;16*(x-5): NEXT x
 750 PRINT AT 19,7;"Time(seconds
)"
 760 PLOT 40,152: DRAW 0,-117: D
RAW 204,0
 770 FOR a=1 TO 100
 780 PLOT 40+2*a,.8*m(a)+36
 790 NEXT a
 800 DEF FN s(t)=256*EXP (-k*t)
 810 INPUT "Guess half life in s
econds ";h
 820 PRINT AT 2,10;"Score=      ";
AT 3,6;"
"
 830 LET point=0
 840 LET k=0.693/h
 850 FOR t=4 TO 400 STEP 4
 860 IF POINT (40+.5*t,.4*FN s(t
)+36) THEN LET point=point+1
 870 PLOT 40+.5*t,.4*FN s(t)+36
 880 NEXT t
 890 DATA "I'm speechless","Plea
se examine your brain","Clever d
ick","You need positive vetting"
 900 PRINT AT 2,16;point;"%";AT
3,6;
 910 FLASH 1
 920 RESTORE 890: READ a$,b$,c$,
d$
 930 IF point=0 THEN PRINT a$
 940 IF point>0 AND point<30 THE
N PRINT b$
 950 IF point>=30 AND point<=60
THEN PRINT c$
 960 IF point>60 AND point<=100
THEN PRINT d$
 970 FLASH 0
 980 PRINT AT 21,0;"Press R to r
un"
1000 IF INKEY$="r" THEN GO TO 10
20
1010 GO TO 1000
1020 PRINT AT 21,0;"Would you li
ke to know more Y/N?"
1030 IF INKEY$="y" THEN GO TO 10
60
1040 IF INKEY$="n" THEN RUN
1050 GO TO 1030
1060 BORDER 0: PAPER 0: CLS : PR
INT PAPER 6; INK 0;"Radioactive
decay simulation         "
1070 PRINT : INK 3: PRINT " Thro
w aside your inhibitions": PRINT
 " and become a nuclear physicis
t!"
1080 PRINT : INK 6: PRINT "Your
research team have amazing,exclu
sive video film of the"
1085 PRINT "radioactive decay of
 256# atoms"
1090 PAUSE 50: PRINT : INK 5: PR
INT "These 'mother' atoms become
 new"
1095 PRINT " !'daughter' atoms"
1100 PRINT : INK 4: PRINT " Use
your observational powers"
1110 PRINT "to deduce the half l
ife: the "
1120 PRINT "time it takes for ha
lf the atoms remaining at any st
age to decay"
1130 PRINT : INK 6: PRINT "A gra
ph of the results plotted"
1140 PRINT "will help but the te
st is severe"
1210 PRINT : PRINT INK 7;"Press
any key to run"
1220 PAUSE 4e4
1230 RUN
```



FERRY

```
  10 LET T=2
  20 LET X=0
  30 LET T=T+1
  40 LET G=25
  50 LET B=4
  60 LET C=B
  70 LET D=C
  80 LET F$="▘"
  90 LET E=25
 100 LET B$="████"
 110 LET C$=B$
 120 LET D$=B$
 125 CLS
 130 IF B=0 AND C=0 AND D=0 AND
T>0 THEN GOTO 30
 135 PRINT TAB 0;T;TAB 14;"█";T
AB 25;X
 140 PRINT AT 1,15;F$;AT 20,3;F$
;AT 20,13;F$;AT 20,23;F$
 150 PRINT AT 21,3;B$( TO B);AT
21,13;C$( TO C);AT 21,23;D$( TO
D)
 165 LET E=INT (RND*50)+1
 170 FOR Z=40 TO 2 STEP -1
 180 IF INKEY$="5" THEN LET E=E-
1
 190 IF INKEY$="8" THEN LET E=E+
1
 200 PLOT E,Z
 210 NEXT Z
 230 IF E=6 THEN LET B=B-1
 240 IF E=26 THEN LET C=C-1
 250 IF E=46 THEN LET D=D-1
 255 IF B=-1 OR C=-1 OR D=-1 THE
N GOTO 325
 260 IF E=6 OR E=26 OR E=46 THEN
 GOTO 263
 262 GOTO 330
 263 LET G=INT (RND*50)+1
 265 FOR Y=2 TO 40
 270 IF INKEY$="5" THEN LET G=G-
1
 280 IF INKEY$="8" THEN LET G=G+
1
 290 PLOT G,Y
 295 NEXT Y
 300 IF G=30 THEN GOTO 315
 310 GOTO 330
 315 LET X=X+1
 320 GOTO 125
 325 IF B=-1 THEN LET B=0
 326 IF C=-1 THEN LET C=0
 327 IF D=-1 THEN LET D=0
 330 PRINT AT 10,16;"CRASH"
 341 LET T=T-1
 343 FOR P=1 TO 50
 344 NEXT P
 346 IF T=0 THEN CLS
 351 IF T=0 THEN PRINT "GAME END
ED. SCORE= ";X
 353 IF T=0 THEN GOTO 1000
 360 GOTO 125
1000 PRINT "AGAIN (Y/N)"
1020 LET O$=INKEY$
1030 IF O$="N" THEN STOP
1040 IF O$="" THEN GOTO 1020
1050 RUN
```

IF THE TEST of a good program is that we cannot tear ourselves away to write the review, Ferry passes. It is not a complicated routine but the cursor keys produce such an instantaneous reaction that the game has a strange hypnotic fascination of its own.

The display shows one port at the top and three at the bottom. One point is awarded for each round trip and an extra ship once all the cargo — 12 loads — is transported successfully. Control the ferry with keys 5 and 8, and do not go back to an emptied quay. Sent by Michael Simmonds for the 16K ZX-81.

Graphics notes:
80 — Graphic 4.
135 — Graphic 8, inverse space.

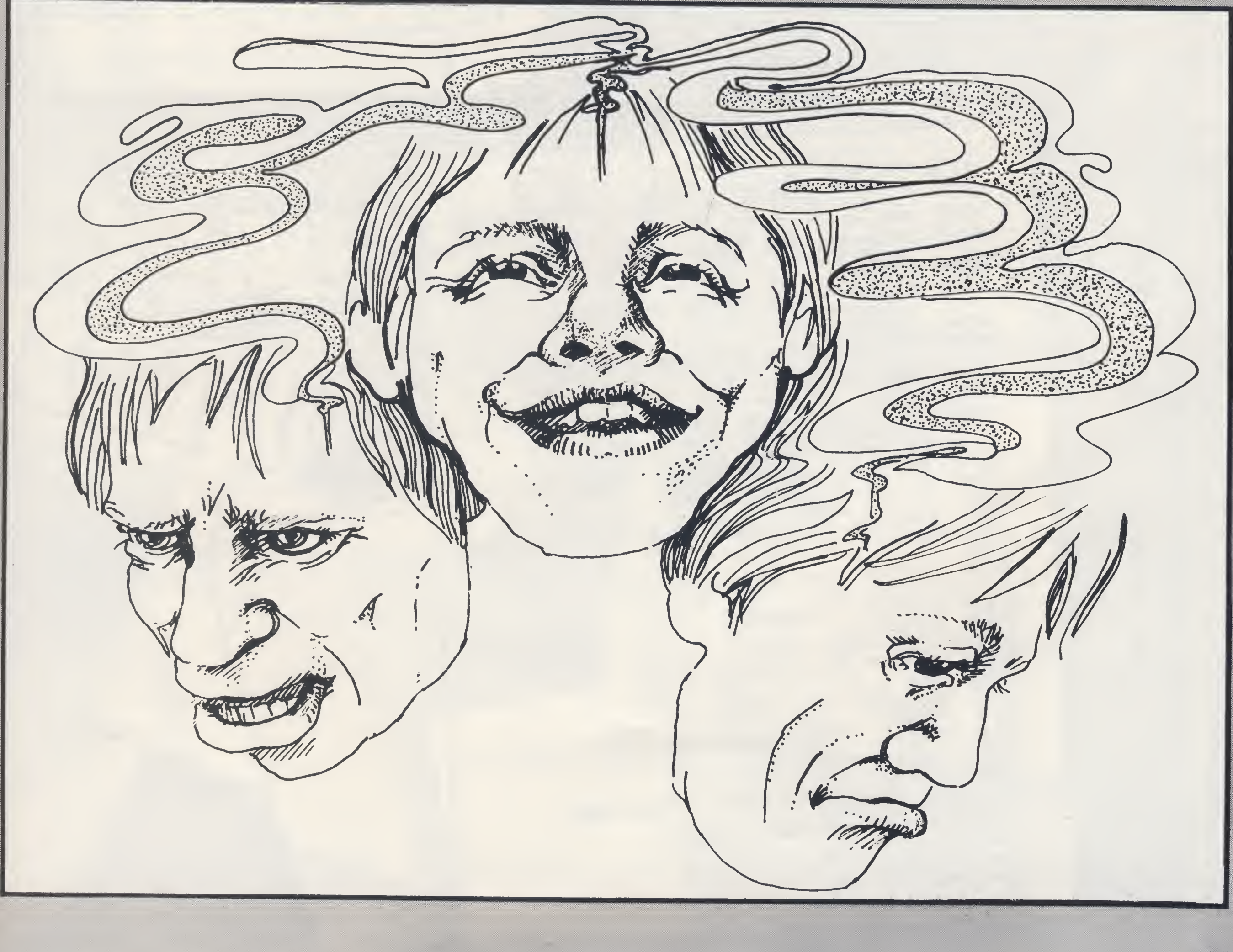
BIO~RHYTHMS

SINCE this program was loaded everybody in the office is becoming a bio-rhythm bore. The program requires your date of birth and the current date and then displays a neat chart of the month, with curves for the physical, mental and emotional cycles.

Your physical state varies over a 23-day cycle and relates to your endurance, strength and aggressiveness. The emotional cycle lasts 28 days and governs anger, moodiness and optimism/pessimism. Mentally, you oscillate between Einstein and ape over a 33-day cycle.

The program was submitted by R Clark of Saltash, Cornwall for the 16K Spectrum.

```
   1 RESTORE
   2 PRINT AT 0,0;"           BIO-R
HYTHM           "
  10 INPUT "Enter Date Of Birth"
,"Day ";a;"  Month ";b;"   Year
";c
  20 INPUT "Enter Date Now ",,"M
onth ";d;"   Year ";e
  25 CLS
  30 LET t=INT (((e-c)*365.25)+(
(d-b)*30.35)-a)
 800 FOR r=0 TO 255
 810 PLOT r,10
 815 IF r=INT (r/8)*8 THEN FOR u
=10 TO 20: PLOT r,u: NEXT u
 820 NEXT r
 830 PRINT AT 21,0;"1st       10th
     20th           30th"
 840 PRINT AT 0,0; INK 1;"physic
al  "; INK 2;"mental   "; INK 4;"
emotional"
 900 FOR r=1 TO 3
 905 READ u
 910 LET l=2*PI*(t-(INT (t/u)*u)
)/u
 920 LET k=2*PI*(33-u)*.03
1000 FOR a=l TO k+l+(2*PI) STEP
.1
1010 PLOT INK ((1 AND u=23)+(2 A
ND u=28)+(4 AND u=33)); (a-l)*(35
-28+u),90+SIN a*60
1020 NEXT a
1030 NEXT r
1040 DATA 23,28,33
1050 INPUT "Another Go ? ";a$: I
F a$(1)="y" THEN GO TO 1
```



THE NUMBER of entries for our May competition proved that as many people are using their ZX-81s for serious uses as for games. The variety of applications was wide and all were of a very high standard.

That made the task of the judges difficult once more but the eventual winner was John Fletcher, of Humberstone, Leicester, who submitted a system which files and retrieves information. He uses it to help keep track of 1,000 photographic slides, with each slide being described by six two-letter codes. It is possible to obtain lists of slides which have up to three codes in common.

Fletcher said he had bought the ZX-81 with the intention of producing such a system. He and his wife had so many slides, which are used to illustrate lectures, that it was difficult to sort them.

He was able to write the program, listed here, after having the machine for only three months.

"I have always been interested in computers and learned how to use the ZX-81 very quickly," he said.

He added that LOADing and SAVEing time was five minutes but that was nothing compared to the time needed previously to sort through the slides.

The system is very general and can be used for other types of filing.

```
   7 FAST
  10 DIM A$(6000,2)
  15 LET Z=1
  20 FOR J=Z TO 1000
  30 FOR K=1 TO 6
  40 PRINT "KEY IN CODES FOR SLI
DE NO.";J
  50 PRINT
  60 PRINT "CODE NO.";K
  70 PRINT
  80 INPUT A$(6*(J-1)+K)
  90 PRINT A$(6*(J-1)+K)
 100 GOSUB 180
 110 IF D$<>"C" THEN GOTO 40
 120 NEXT K
 130 PRINT "ARE THERE ANY MORE S
LIDES? Y/N"
 140 INPUT D$
 150 CLS
 160 IF D$="Y" THEN NEXT J
 165 LET Z=J+1
 170 GOTO 300
 180 PRINT AT 15,1;"KEY C IF THI
S IS OK. OTHERWISE NEWLINE"
 190 INPUT D$
 200 CLS
 210 RETURN
 220 FOR K=1 TO 6
 230 IF A$(6*(J-1)+K)=F$ THEN PR
INT J;" ";
 240 PRINT
 250 NEXT K
 260 RETURN
 300 PRINT ..[illegible]
[illegible]"
 320 PRINT
 330 PRINT "FOR INDIVIDUAL SLIDE
S     KEY 0"
 340 PRINT
 350 PRINT "FOR 1-TERM SELECTION
      KEY 1"
 360 PRINT
 370 PRINT "FOR 2-TERM SELECTION
      KEY 2"
 380 PRINT
 390 PRINT "FOR 3-TERM SELECTION
      KEY 3"
 400 PRINT
 410 PRINT "TO ADD MORE SLIDES
      KEY 4"
 420 PRINT
 422 PRINT "TO SAVE PROGRAM
      KEY 5"
 424 PRINT
 430 PRINT "TO CHANGE PROGRAM
      KEY 9"
 440 INPUT A
 450 CLS
 460 IF A=0 THEN GOTO 500
 470 IF A=1 THEN GOTO 660
 480 IF A=4 THEN GOTO 1300
 485 IF A=5 THEN GOTO 1500
 490 IF A=9 THEN LIST
 495 GOTO 300
 500 PRINT "KEY IN SLIDE NO."
 510 INPUT J
 520 CLS
 530 PRINT "SLIDE NO.";J;" HAS T
HE FOLLOWING"
 540 PRINT "CODES:"
 550 PRINT
 560 FOR K=1 TO 6
 570 PRINT A$(6*(J-1)+K)
 580 NEXT K
 590 PRINT
 600 PRINT
 610 PRINT "ANOTHER SLIDE? Y/N"
 620 INPUT D$
 630 CLS
 640 IF D$="Y" THEN GOTO 500
 650 GOTO 500
 660 PRINT "ENTER CODE"
 670 INPUT F$
 680 PRINT
 690 PRINT F$
 700 GOSUB 180
```

FILIN

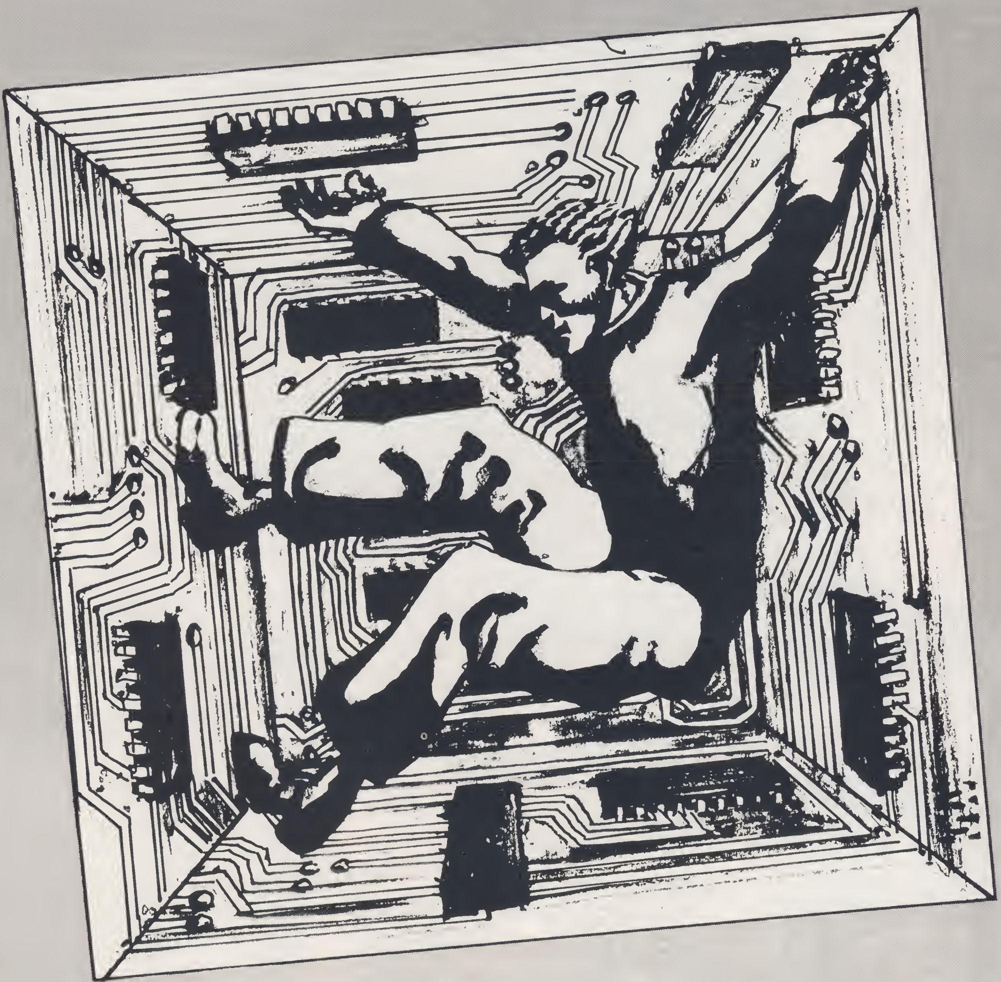
-WS-

```
710 IF D$<>"C" THEN GOTO 660
720 PRINT "LIST OF SLIDES HAVIN
G CODE ";F$
730 PRINT
740 FOR J=1 TO Z
750 GOSUB 220
760 NEXT J
770 GOTO 1200
800 PRINT "ENTER FIRST CODE "
810 INPUT F$
820 PRINT
830 PRINT F$
840 GOSUB 180
850 IF D$<>"C" THEN GOTO 800
860 PRINT "ENTER SECOND CODE"
870 INPUT G$
880 PRINT
890 PRINT G$
900 GOSUB 180
910 IF D$<>"C" THEN GOTO 860
920 IF A=2 THEN GOTO 990
930 PRINT "ENTER THIRD CODE"
940 INPUT H$
950 PRINT
960 PRINT H$
970 GOSUB 180
980 IF D$<>"C" THEN GOTO 930
990 PRINT "LIST OF SLIDES HAVIN
G FOLLOWING CODES:"
1000 PRINT
1010 PRINT F$;", ";G$;
1020 IF A=3 THEN PRINT " AND ";H
$
1030 PRINT
1040 PRINT
1050 IF A=3 THEN GOTO 1130
1060 FOR J=1 TO Z
1070 FOR L=1 TO 6
1080 IF A$(6*(J-1)+L)=G$ THEN GO
SUB 220
1090 IF A$(6*(J-1)+L)=G$ THEN NE
XT J
1110 NEXT L
1120 GOTO 1200
```

SYSTEM

```
1130 FOR J=1 TO Z
1140 FOR M=1 TO 6
1150 IF A$(6*(J-1)+M)=H$ THEN GO
TO 1070
1160 NEXT M
1170 NEXT J
1200 PRINT
1210 PRINT
1220 PRINT
1230 IF A=1 THEN PRINT "ANOTHER
CODE? Y/N"
1240 IF A<>1 THEN PRINT "ANOTHER
 SET OF CODES? Y/N"
1250 INPUT D$
1260 CLS
1270 IF D$<>"Y" THEN GOTO 300
1280 IF D$="Y" AND A=1 THEN GOTO
 660
1290 GOTO 800
1300 LET J=Z-1
1310 PRINT "LAST SLIDE ADDED WAS
 NO.";J
1320 PRINT "SLIDE NO.";J;" HAS T
HE FOLLOWING"
1330 PRINT "CODES:"
1340 PRINT
1350 FOR K=1 TO 6
1360 PRINT A$(6*(J-1)+K)
1370 NEXT K
1380 PRINT
1390 PRINT
1400 PRINT "PRESSN/L TO ADD MORE
 SLIDES"
1410 INPUT D$
1415 CLS
1420 GOTO 20
1500 CLS
1510 PRINT AT 10,10;"START TAPE"
1520 PRINT TAB 10;"KEY N/L"
1530 INPUT D$
1540 CLS
1550 LET D$="SLIDES 1"
1560 SAVE D$
1570 GOTO 300
```

ENGULF



ENGULF is for the 1K ZX-80, by Graham Charlton of Rise Park, Romford, Essex. It sets you against a vengeful computer. After you press RUN, a large black square will appear. That is the playing area. You have five controls:

7 — UP
6 — DOWN
5 — LEFT
8 — RIGHT
9 — SCORE

After pressing 5, 6, 7 or 8, followed by NEWLINE, you will appear as an addition sign (+) with an inverse space, CHR$(128), next to it. Every time you move, the computer will place a black square next to your piece. The idea is to avoid being trapped, or engulfed, for as long as possible. Once you have been caught, press 9 to see your final score.

Charlton's best score is 154. You can make the game slightly easier by deleting line 280, which prevents the computer placing a black square where a black square already exists. The game shows PEEKing and POKEing to the display file in action. The crucial line is 100, which finds the beginning of the display file.

```
10 LET A = 0
20 LET E = 236
30 FOR C = 1 TO 21
40 FOR G = 1 TO 21
50 PRINT "*";
60 NEXT G
70 PRINT
80 NEXT C
90 LET G = 0
100 LET A = 1 + PEEK (16396) +
   PEEK (16397)*256
110 FOR C = 0 TO 20
120 POKE A + C, 128
130 POKE A + C + 22*20, 128
140 NEXT C
150 FOR C = 1 TO 21
160 POKE A + 22*C, 128
170 POKE A + 22*C + 20, 128
180 NEXT C
190 INPUT D
195 IF D>9 OR D <5 THEN GO TO
   190
200 GOSUB 320
210 IF D = 9 THEN GO TO 380
220 IF PEEK (A + F) = 128 THEN
   GO TO 190
230 POKE A + E, 0
240 LET E = F
250 POKE A + E, 19
260 LET D = RND(4) + 4
270 GOSUB 320
280 IF PEEK (A + F) = 128 THEN
   GO TO 260
290 POKE A + F, 128
300 LET G = G + 1
310 GO TO 190
320 LET F = E
330 IF D = 5 THEN LET F = F-1
340 IF D = 8 THEN LET F = F + 1
350 IF D = 6 THEN LET F = F + 22
360 IF D = 7 THEN LET F = F-22
370 RETURN
380 PRINT "YOU SCORED*";G
```

HELPLINE

The ZX-80, ZX-81 and Spectrum have introduced many people to the world of computers. Part of the interest has been in learning to use the techniques available. That has often given rise to many problems and queries. Our Helpline service was set up to deal with them and has proved very popular. We asked our correspondent, Andrew Hewson, to select some of the more important items from his postbag.

Sinclair memories are made of this

PROBLEMS which individual Sinclair users experience can be helpful to others if they are answered explicitly. Here is a representative selection.

"Please explain the meaning of an "address". How can a byte, which is a number, have an address?"

An important part of a computer is its memory and typical microcomputers have several thousand memory locations available for immediate use. Clearly each location needs a separate label or address to distinguish it from its fellows. The word address came into use because writing information to one of many memory locations is similar to writing a letter to one of many people. Letters are sent to an address so that they reach the person who lives there. Similarly, a computer sends information to the memory location at a given address.

Computer addresses are simply whole numbers starting at zero; so, for example, in the unexpanded XZ-81, locations 0 to 8191 are used by the ROM, locations 8192 to 16383 are unused, and locations 16384 to 17407 are used by the RAM. The add-on 16K RAM uses locations 16384 to 32767. Only the contents of RAM may be altered and so users generally are interested in addresses 16384 and upwards.

Each location in memory contains one byte of information. A byte can be thought of as a whole number between 0 and 255 inclusive. In practice, the word byte is often also used to mean a "location in memory" as well as to mean the number which is stored at that location. Thus, if location 17000 contains 34, we might say "byte 17000 is 34".

"I am keen to understand how my ZX-81 works but as a beginner I am perplexed by the manner in which addresses are stored in the system variables. I know, for example, that D-FILE is the begining of the display file — but how is that information stored?"

The area at the bottom of RAM between 16384 and 16508 holds the system variables and is followed by the program area, starting at 16509. The display file is next but as programs can vary in length, the display file does not start at a fixed address. The ZX-81 keeps track of it by storing the current value of the starting address in D-FILE.

If you look at page 178 of the manual you will see that the value of D-FILE is stored at address 16396 and so you might infer that you have only to look at the contents of 16396 to find the value of D-FILE.

Unfortunately, that is not true. Remember that the value of D-FILE is an address and that addresses are whole numbers, like 16384 and 17407 and 32767. A single location can hold only a number between 0 and 255 and so two adjacent locations are used to store larger numbers. The value of D-FILE given by:

value held in 16397 + 256*
value held in 16397

Any whole number between 0 and 65535 inclusive can be stored using this system.

The value held at an address can be found by PEEKing it and so you can PRINT the value of D-FILE by entering:

```
PRINT PEEK 16396 + 256*PEEK
16397
```

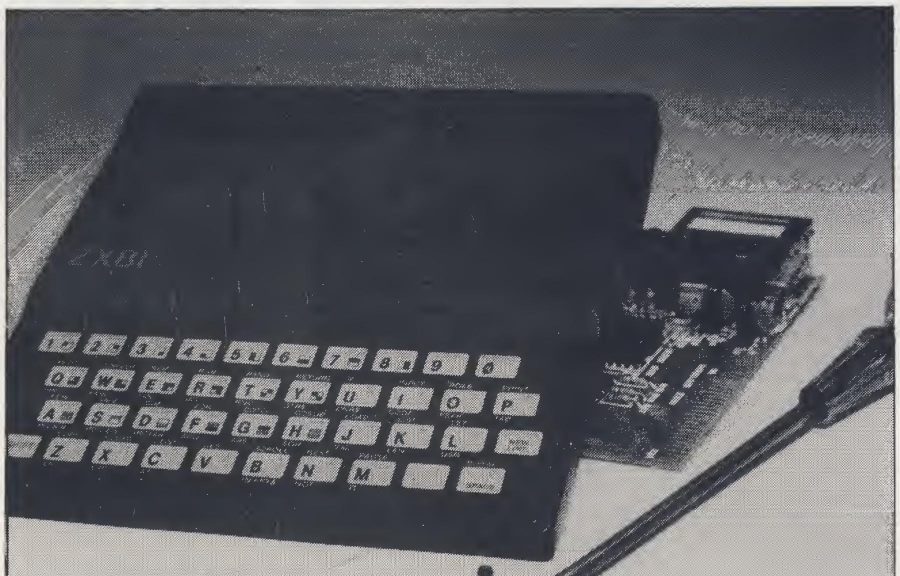
You may know that the contents of the first location in the display file is always 118 and you can show that by entering:

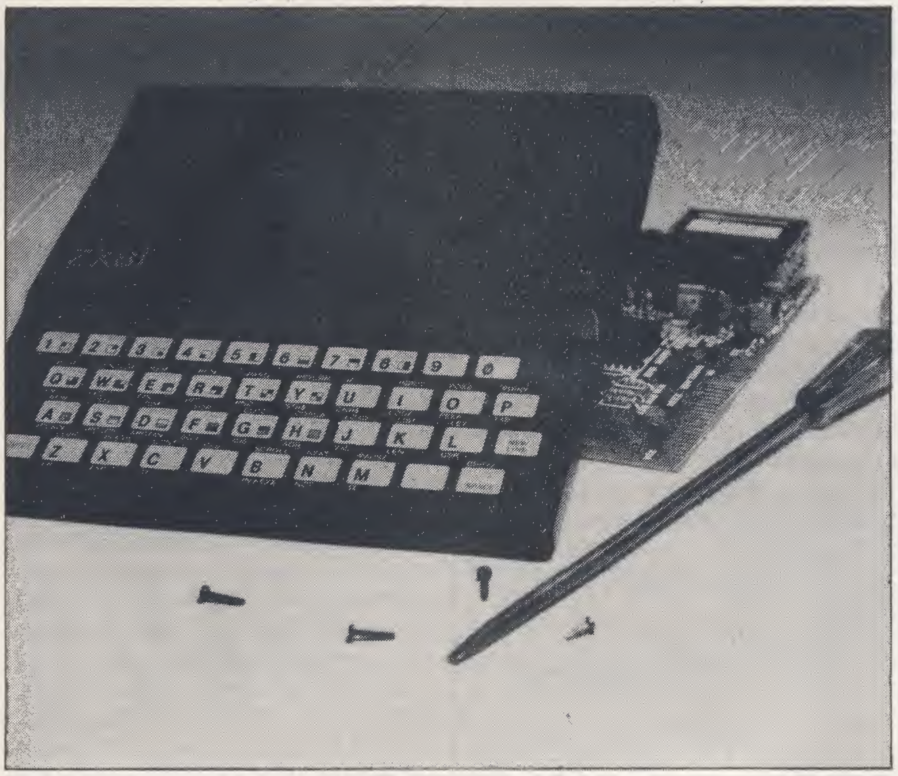
```
PRINT PEEK (PEEK 16396 + 256*
PEEK 16397)
```

"I want to write several programs which use the same data but there seems to be no way to do so using the Sinclair functions. Do I have to write routines to save and load data on cassette or is there some way of passing data between programs directly?

There is. The trick is to alter the RAMTOP system variable to give you some space at the top of RAM which is out of reach of the Sinclair system in normal use. Your first program can then copy data into the area above RAMTOP. You can then load a second program, replacing the first, but the data saved above RAMTOP will still be intact. The second program can then copy the data back into its own variables area.

Let us take the job step by step. All the calculations following refer to the unexpanded ZX-81, with values for the 16K expansion in square brackets where they are different.





Moving RAMTOP. Suppose you want to pass a 10-element array between programs. A single-dimensional array occupies five bytes per element plus another six, making 56 bytes in all, and so you need at least 56 bytes above RAMTOP.

When you switch on your ZX-81, RAMTOP os set automatically to one more than the top of RAM, i.e., 17408 [32768 with the 16K RAM]. The address is stored as a system variable at 16388 and 16389 as the values 0 and 68 respectively [0 and 128], because 0 + 256 *68 = 17408 [0 + 256*128 = 32768].

You can use your ZX-81 to calculate the new value to be POKED into 16388 by entering:

```
PRINT 17408-N-256 *INT ((17408-
                N)/256)
[PRINT 32768-N-256*INT ((32768-
                N)/256)]
```

Using N = 56 for our example gives 200.

The value to be POKEd into 16389 is given by:

```
    PRINT INT ((17408-N)/256)
   [PRINT INT ((32768-N)/256)]
```

In our example the result is 67 [128].

The ZX-81 will ignore any alterations to RAMTOP until you enter NEW, so do so at that point. Of course, if you have a program in your machine you should SAVE it first.

Finding your array. Variables are stored at the address held in VARs at 16400 and 16401 so you can PRINT the current value by entering:

```
 PRINT PEEK 16400 + 256 * PEEK
                16401
```

Saving an array above RAMTOP. The following program demonstrates the technique:

```
10 DIM A(10)
20 FOR I = 1 TO 10
30 LET A(I) = I
40 NEXT I
50 LET J = PEEK 16400 + 256 *PEEK
16401
60 LET K = PEEK 16388 + 256*PEEK
16389
70 FOR I = 0 TO 55
80 POKE K + I, PEEK (J + I)
90 NEXT I
```

Lines 10 to 40 assign the array and set its values to 1. . .10 — those values have been chosen for the purposes of this demonstration and have no special significance. Line 50 stores the value of VARS in J and line 60 stores the value of RAMTOP in K. The loop at lines 70 to 90 copies the contents of the array above RAMTOP.

Retrieving an array from above RAMTOP. Now delete the first program and enter the following:

```
10 DIM A(10)
50 LET J = PEEK 16400 + 256*PEEK
16389
70 FOR I = 0 TO 55
80 POKE J + I, PEEK (K + I)
90 NEXT I
100 FOR I = 1 TO 10
110 PRINT A(I)
120 NEXT I
```

In this program the array is assigned and J and K are set to VARS and RAMTOP as before but the loop at lines 50 to 780 now copies the data from above RAMTOP to the variables area. The loop at lines 100 to 120 PRINTs the values of the array as set by the first program.

In each case, lines 50 to 90 represent the essential part of the program but it is important to assign the array at the beginning of the program, so that it lies at the bottom of the variables area. The same technique works for ordinary variables but strings cannot, in general, be copied in this manner, because they can move around in RAM.

"Deleting lines by entering their line numbers in turn is a nuisance. Is there a method for deleting blocks of lines?"

There is, because the length in bytes of each line is held in two locations immediately following the line number and we can manipulate them to our advantage. To see the line length, clear your ZX-81 by entering NEW and then enter the following line:

```
10 REM
```

Then PEEK at the first few bytes in the program area which starts at 16509 and you will see:

Address	Contents	Meaning
16509	0	line number =
16510	10	256*o + 10 = 10
16511	2	line length =
16512	0	2 + 256*0 = 2
16513	234	code for REM
16514	118	code for newline

Notice that the line length excludes the four bytes required for the line number and the line length. The following routine finds the address of the first byte of two lines entered by the user:

```
9000 DIM B(2)
9010 LET START = 16509
9020 PRINT "ENTER FIRST AND
LAST LINE NOS"
9030 INPUT B (1)
9040 INPUT B (2)
9050 IF B (1)<0 OR B(2)<0
OR B (1)>9999 OR
B(2)> 9999 THEN GOTO 9020
9100 FOR I = 1 TO 2
9110 LET L = PEEK (START + 2)
 +256*PEEK (START + 3)
9120 LET LINEND = 256*PEEK
```

START + PEEK (START + 1)
9130 LET START = START + L + 4
9140 IF LINEND <B (I) THEN GOTO 9110
9150 LET B(1) = START — (L + 4)*(I = 1)
9160 NEXT I

The routine finishes with the two addresses held in B(1) and B(2). The next trick is to fool the ZX-81 into thinking that all the lines between and including the two line numbers are just one monster line by POKE-ing the difference between B (2) and B (1) less 4 into the slot for the line length of the first line. That is achieved by adding the lines:

9200 LET B(2) = B(2) — B(1) — 4
9210 POKE B(1) + 2, (B(2) — 256*INT (B(2)/256)
9220 POKE B(1) + 3, INT (B(2)/256)

The monster line can then be deleted in the usual way, by entering the line number alone.

"I would like to know how to SAVE and LOAD data only into a program held in core. Can you assist?"

There are two methods. The first is quick and elegant and consists essentially of writing new SAVE and LOAD routines in machine code. The second method is slow and clumsy but it is easy to understand and the necessary software is mostly in Basic, the preferential method. The steps in outline are:

SAVE the data of interest on tape; to prevent the program in the ZX-81 being over-written, copy the program area above RAMTOP; LOAD the data from tape in the usual way; create some space in the program area and copy the program from above RAMTOP into the newly-created space in the program area.

Obviously, to create some data to LOAD into a program we must RUN a previous program to read in or calculate the data to be SAVEd. The earlier program could be deleted line by line but the process is rather laborious, so use this technique instead:

First note the line number of the first line of the program. Suppose it is line number 10; then find the effective length of the program by entering PRINT PEEK 16397 — 16513.

Suppose the result is 1859. Then enter POKE 16511, 1859 — 256* INT (1859/256); POKE 16512, INT (1859/256); 10 or whatever was the first line number.

Do not attempt to LIST the program between entering the instructions or you will have to pull out the plug and start again.

The data can then be SAVEd, together with the display file and other odds and ends, on tape.

The second step is to LOAD a new program and store it above RAMTOP. The technique is very similar to storing data above RAMTOP. Be sure to move RAMTOP down as explained on page 168 of *ZX-81 Basic Programming* before LOADing the new program. The following routine copies a program above RAMTOP:

10 LET J = PEEK 16396 + 256*PEEK 16397 — 16509
20 PRINT J
30 LET K = PEEK 16388 + 256*PEEK 16389
40 FOR I = 0 to J—1
50 POKE K + I, PEEK (16509 + I)
60 NEXT I

The routine PRINTs the length of the program, J, in bytes. You should note it as it will be needed later.

The data can then be LOADed from tape in the usual way. The current program will, of course, be over-written and so the final step is to copy it back from above RAMTOP. A machine code routine is needed for that step, because space must be created in the program area in which to store the program using a routine in ROM.

The routine is 20 bytes long and you can store it at addresses 32748 to 32767 by entering and RUNning the following routine:

10 FOR & = 32748 to 32767
20 INPUT M
30 POKE I, M
40 PRINT I, PEEK I
50 NEXT I

Enter the following numbers one by one from the keyboard: 42, 12, 64, 229, 43, 1, 0, 0, 197, 205, 158, 9, 193, 209, 42, 4, 64, 237, 176, 201. You might like to determine how the routine works by tranlating the decimal numbers into Z-80 assembler using Appendix A of the *ZX-81 Basic Programming* manual.

Before running the machine code routine, POKE the program length, J, into it by entering

POKE 32754, J — 256*INT (J/256)
POKE 32755, INT (J/256)

Then delete the Basic routine, put the ZX-81 into FAST mode, and call the machine code routine by entering
IF USE 32748 = 0 THEN STOP.

Numbers games

STORING and displaying numbers is the object of many requests for information. The ZX-81 uses one of three methods, depending on the context. The first is the floating point method which is used for all Basic variables and all calculations involving Basic variables. All numbers held using the technique occupy five bytes each. The second method is used internally by the Z-80 microprocessor which drives the ZX-81 and can be used for whole numbers only.

Each number occupies only two bytes and a variation of the method is used to store the line numbers at the beginning of each Basic program line. To communicate to the user the ZX-81 uses a third method, the character form, in which each decimal digit, and the decimal point, occupies one byte each. Some typical queries:

"What does floating point arithmetic mean? Why use it? Would it not be easier to use the decimal system?

The use of floating point arithmetic implies that numbers are stored and manipulated as a mantissa, which contains the digits in the number, and an exponent, which indicates the position of the decimal point. It is relatively easy to convert decimal numbers into a decimal floating point representation and some calculators can display decimal numbers in that form. The calculator manufacturers refer to the form as scientific notation.

The great advantage of scientific notation is that very large numbers, or numbers which are very close to zero, can be written approximately using a limited number of digits. Thus a calculator which can display only eight digits at a time can display a number larger than 99,999,999 using scientific instead of ordinary notation.

For example, the distance from the Earth to the Sun is about 5,892,480,000,000 inches. Thirteen digits are required to write the number in ordinary notation but

when it is re-written in its scientific form as 5.89248×10^{12} only eight digits are required — neglecting the $\times 10$ which is common to all numbers written in that way. The mantissa is 589248 and the exponent is 12.

The exponent, by the way, means "imagine that the decimal point is to the right of the left-most digit of the mantissa — i.e., between the 5 and the 8. Now move the point 12 places to the right, filling spaces with zeros if necessary".

The scientific form, of course, is not accurate because only eight digits, of which six only are in the mantissa, are allowed, whereas in the ordinary form 13 are available, although in our example the extra digits are all zeros.

We count in tens and so calculators display numbers in decimal for our convenience. Digital computers count in binary as explained in chapter 24 of the *ZX-81 Basic programming* manual but the principle of floating point binary representation is the same as that of decimal scientific notation. The ZX-81 uses a string of seven zeros and ones for the exponent — i.e., one byte with one bit reserved for the sign of the exponent — and a string of 31 zeros and ones for the mantissa — four bytes with one bit used for the sign of the mantissa.

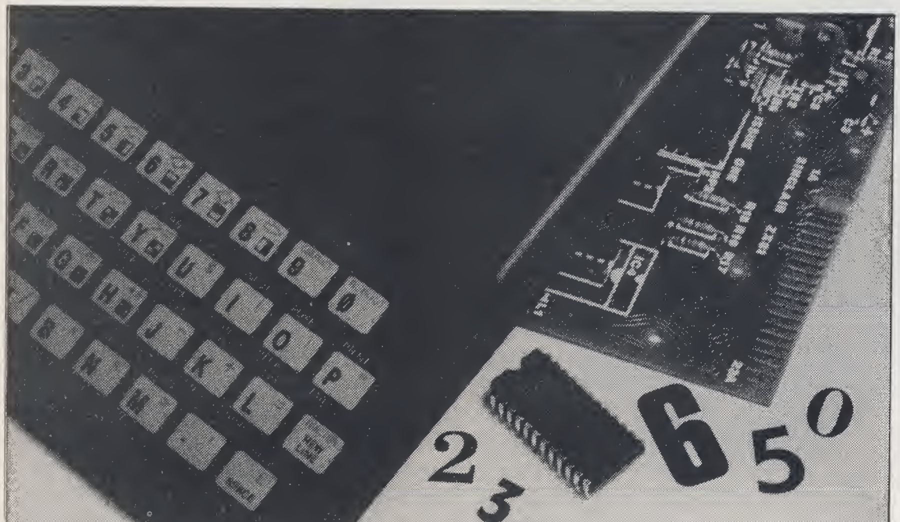
Floating point arithmetic is used because it enables a large range of numbers to be stored in five bytes with only a small loss in accuracy. Numbers as big as 10^{38} — one followed by 38 zeros — can be stored, although only the first nine digits or so are accurate. If integer arithmetic were used, then the biggest number which could be stored in the same space would be 1,099,511,627,776 but all 13 digits would be accurate.

"How are numbers stored in the ZX-81? Please explain how the five-byte representation of a number is obtained".

The following program prints the floating point form of a number entered by the user at line 20:

```
10 PRINT "ENTER A NUMBER"
20 INPUT I
30 PRINT,, "THE ZX81
REPRESENTS": I: "BY"
40 FOR J = 1 TO 5
50 PRINT PEEK (PEEK 16400 + 256*
PEEK 16401 + J); "  ";
60 NEXT J
70 PRINT
80 PAUSE 500
90 CLS
100 RUN
```

The ZX-81 stores the values of all Basic variables in the variables areas and the address of the beginning of the variables area is held in VARS at locations 16400 and 16401 — see chapter 28 of the ZX-81 manual. Thus the loop at lines 40 to 60 prints the contents of the five bytes which hold the floating point version of the number entered.



The first of the five bytes is the exponent, E, and the remaining four bytes, A, B, C, D represent the mantissa. If the original number is positive, A lies in the range 0 to 127. If it is negative, A lies between 128 and 255.

The following program re-contructs a number from its Sinclair floating point form:

```
10 PRINT "ENTER THE
EXPONENT AND THE FOUR
NUMBERS OF THE MANTISSA,
ALL ENTRIES TO LIE BETWEEN 0
  and 255 INCLUSIVE"
20 INPUT E
30 INPUT A
40 INPUT B
50 INPUT C
60 INPUT D
70 INPUT,, "EXPONENT = "
,E,"MANTISSA = "', A,,B,,C,,D
80 PRINT,, "THE NUMBER = ";(2*
(A<128)-1)*2**(E-160)*(((256*
(A + 128*(A<128)) + B)*256 + C)
*256 + D)
```

Try those two programs for a variety of numbers. You will see that the exponent is about 128 for numbers close to 1 and -1; that numbers close to 0 have small exponents; and that large positive and large negative numbers have large exponents.

It is also noticeable that the value of the fourth byte, D, has little or no effect on the value printed by the second program. In computer jargon D is called the least significant byte. The ZX-81 prints results to eight decimal figures at most, rounding the result if necessary, although calculations are made to somewhat greater accuracy.

"I am writing a program to test my son's arithmetic but I have found that my ZX-81 sometimes gets the answer wrong. I know that some early ZX-81 ROMs made an error with the square of 0.25 but my machine does not make that mistake. Is this another bug?"

The fault lays with the program and not with the ZX-81. The routine which was at fault set a problem in division and then compared the user's reply to the result calculated by the ZX-81. Unfortunately the routine often rejected correct replies.

It is often impossible to convert a

decimal number exactly to a binary floating point number and that was the source of the problem. An analogous difficulty can occur when converting some fractions into decimal — we are all familiar with the fact that ⅓ cannot be written as an exact decimal. The program was rejecting the user's reply even when it differed by only a tiny amount from the calculated result.

The following program asks you to enter a number, divide it by 10 and enter the result. It then prints the floating point representation of your result and its own result for the same calculation. If you run the program a few times you will see that your answer and the answer produced by the ZX-81 often differ by one in the least significant digit of the mantissa.

```
10 LET N = 500
20 CLS
30 PRINT "ENTER A NUMBER"
40 INPUT I
50 PRINT,, "YOU ENTERED"; I
60 PRINT,, "DIVIDE"; I; BY 10"
70 PRINT "ENTER THE RESULT"
80 INPUT J
90 PRINT,, "YOU ENTERED"; J
100 LET K = I/10
110 IF ABS (K-J)<.0001*K THEN
GOTO 170
120 PRINT,, "WRONG"
130 PRINT I; "DIVIDED BY 10 DOES
NOT EQUAL";J
140 PRINT "TRY AGAIN"
150 PAUSE N
160 RUN
170 PRINT,, "RIGHT"
180 RPINT I; "/10 = ";J
190 PAUSE N
200 CLS
210 PRINT,, "THE ZX81
REPRESENTS";J;"BY"
220 LEt M = 13
230 GOSUB 300
240 PRINT,, "AND";I;"/10 BY"
250 LET M = 19
260 GOSUB 300
270 PAUSE N
280 RUN
300 FOR L = M TO M + 4
310 PRINT PEEK (PEEK 16400 + 256
*PEEK 16401 + L);" ";
320 NEXT L
330 PRINT
340 RETURN
```

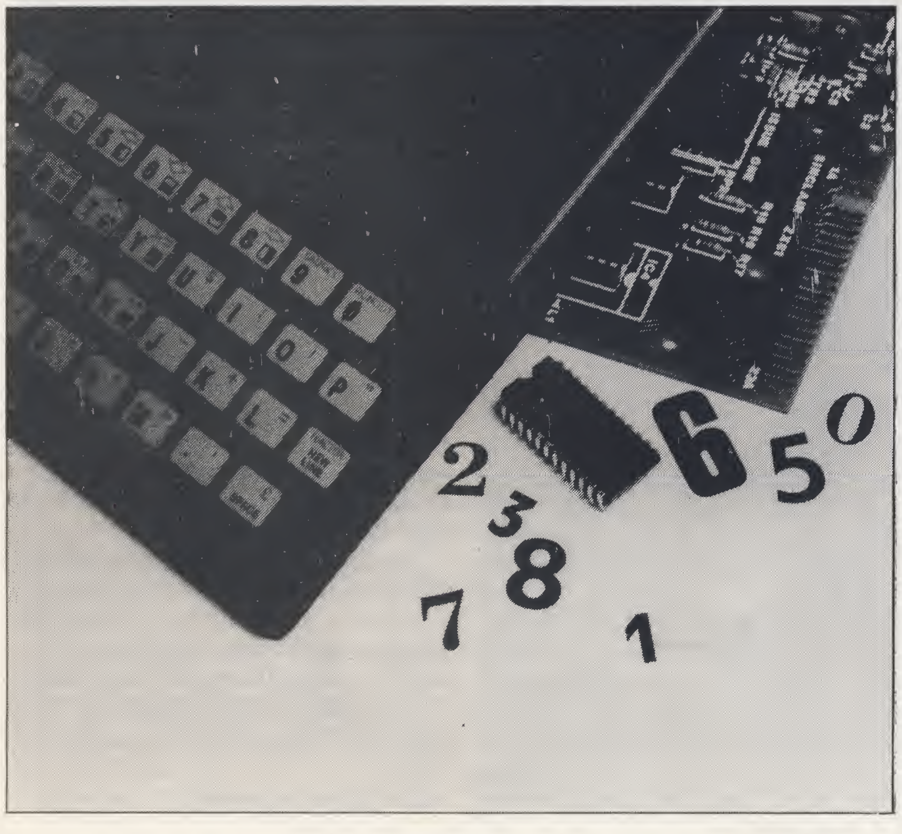
If you wish to avoid problems of that nature then you should alter statements like

IF K = J THEN GOTO 170

to IF ABS (K-J)<.0001*K THEN GOTO 180

In the first case the program will jump to line 170 only if K and J are identical down to the last digit. In the second case, the jump will be made if the difference between K and J is less than .01 percent.

"If the ZX-81 uses two bytes to store line numbers, why is 9999 the largest line number permitted?"



Each byte contains eight bits and each bit can take two values giving $2^{16} = 65536$ arrangements of the 16 bits in the two bytes. Hence the two bytes could be used to represent any positive integer between 0 and 65535 inclusive. Why limit line numbers to 9999?

The reason is that by limiting in this way and by manipulating the numeric codes for variables the ZX-81 has a device for distinguishing lines in the program area from variables in the variables area.

To understand the mechanism at work, consider the representation of 9999. Line numbers are held with their most significant byte first, contrary to the usual Z-80 convention, so that line number 9999 is held as a byte containing 39 followed by a byte containing 15 because 39*256 + 15 = 9999. The bit pattern of the first byte, obtained by converting 39 to binary, is 00100111. Notice that the three most significant bits — bit numbers 7, 6 and 5 are set to 0, 0 and 1 for this, the largest permitted line number. Hence bit numbers 7, 6 and 5 of the first byte of all permitted line numbers will be set to 0, 0 and 1, or in the case of line numbers less than 8192 they will be set to 0, 0, 0.

Now look at pages 172 to 174 of the *ZX-81 Basic Programming* manual and you will see illustrations of the different types of variables as they are represented in the variables area. In each case the first byte contains a numeric code related to the code of the letter; in the case of a number whose name is longer than one letter, the first letter which identifies the variable.

The largest possible letter code is 63, the code for Z, which is 00111111 in binary, and the smallest is 38, the code for A, which is 00100110 in binary. Clearly, bits 7 and 6 are not needed when distinguishing between letter codes and bit 5 is always set to one, so the ZX-81 uses them to distinguish between the different types of variable, subtracting 20h, or 32 in decimal, from the letter code in three of the six cases.

Three bits can be set in 2^3, or 8, different ways. The table lists the eight ways and their interpretation:

Bit pattern	Interpretation
000	Line number less than 8192
001	Line number between 8192 and 9999

010	String
011	Number with single character name
100	Array of numbers
101	Number with multiple character name
110	Character array
111	Control variable for a FOR-NEXT loop

The same purpose could be served by comparing the address of the byte in question to the D-FILE or VARS pointers. It allows the ZX-81 to use the same routine, at 2546 to 2576, to step through memory to the "next" line or the "next" variable but that seems a small advantage.

Perhaps it is merely a hangover from the ZX-80, because in that machine the variables area follows immediately after the program area and so a device which "knows" from the contents of the byte that the end of the program has been reached serves some purpose.

It is worth noting that while the ZX-81 prevents you entering line numbers greater than 9999 from the keyboard, if you manipulate the line numbers by POKEing the appropriate locations your program will still run, provided the line numbers do not exceed 16383 as the following routine demonstrates:

```
  10 LET I = 10000
  20 SCROLL
  30 PRINT I
  40 POKE 16634, INT (I/256)
  50 POKE 16635, I–256*INT
     (I/256)
  60 LET I = I + 1
9999 GOTO 20
```

Line numbers 40 and 50 POKE the current value of I into the locations originally occupied by 9999. If you run the program for a few cycles and then BREAK it you will see that 9999 has been updated to, say, AO29 for I = 10029.

Clearly the ZX-81 does not decode line numbers greater than 9999 correctly but the result is comprehensible if you remember A follows 9 in the sequence of character codes.

If you leave the program running for long enough it will stop when I = 16384 and a LISTing will then omit the final line because the LIST command does not recognise it as a line.

You can use this quirk to make programs "disappear" by POKE 16509, 64. Such "invisible" programs can be SAVEd and LOADed as usual and will RUN if 16509 is re-set to its original value.

File problems on display in ZX-81

THE display file is the area in RAM which holds the items which are currently displayed on the TV screen. During program development, for example, part of the program listing is generally displayed and it is the job of the LIST routine to copy the appropriate part of the program area into the display file so that it appears on the TV screen.

The address of the beginning of the display file varies with the length of the program and so it is held in the D-FILE system variable and can be PRINTed by entering:
PRINT PEEK 16396 + 256*
PEEK 16397

Similarly, the address of the end of the display is held in VARS — or more correctly is one less than the value in VARS — and can be PRINTed by entering:
PRINT PEEK 16400 + 256*
PEEK 16401-1

Each line displayed on the screen is terminated by a byte containing decimal 118 and there is one more byte at the beginning of the file also containing 118. As there are 24 lines in the display, there are 25 bytes containing 118 and it is essential to the correct functioning of the display software that all 25 bytes are present. If one or more is absent the ZX-81 will almost certainly crash.

With an unexpanded ZX-81 the display file consists of those 25 bytes and no more unless a LIST, PRINT or PLOT command has been executed since the last CLS or RUN command. That is a device to keep the display file as small as possible, thereby saving valuable space in RAM. With a memory expansion pack of 4K or more the ZX-81 pads-out the display file with 32 bytes per line each containing zero — i.e., blanks. That mechanism creates an immediate problem for users without extra memory.

"I am having problems with programs which POKE the display because my ZX-81 crashes every time. For instance
*5 LET Z = 1 + PEEK 16396 + 256**
POKE 16397
10 POKE Z, 128
is a disaster. What is happening?"

The routine is attempting to POKE an inverse space into the first PRINT position and it will work well on a ZX-81 with at least 4K of RAM because the display will be padded-out with 32 blanks per line. Without the extra memory, no such padding occurs and so the routine will over-write the second of the 25 bytes containing 118, thus causing the program to crash.

The solution to the problem and all similar ones is to create some space at the appropriate position in the display file by PRINTing one or more blank characters. In that case one space is required at the begining of the first line and so the answer is to add to line:

```
2 PRINT "b"
```

where b represents the space character.

"I would like to give a print instruction conditional on whether or not there is anything already printed at the same position. I solved the corresponding problem by using PEEK 16438 and PEEK 16439 but the use of PEEK 16441 and PEEK 16442 does not work. Can you help?"

Page 179 of the *ZX-81 Basic Programming* manual shows that addresses 16438 and 16439 in the system variables hold the x and y co-ordinates of the last point to be PLOTed. Unfortunately the equivalent information for PRINT is not stored. Instead the position at which the next item will be PRINTed is held in 16441 and 16442. There is a further complication; the horizontal PRINT position is counted from right to left and the vertical position from the bottom upwards. Thus the routine:

```
10  PRINT "HELP-LINE"
20  LET A = PEEK 16441
30  LET B = PEEK 16442
40 PRINT A, B
```

will print the values 33 23. The first value, 33, indicates that the PRINT position is at the beginning of a line, i.e., 33 characters counting from right to left from the beginning of of the subsequent line. The second value, 23, indicates that the line is the 23rd from the bottom of the screen because one line has been used to PRINT "HELP-LINE". If line 10 is changed to

```
10  PRINT "HELP-LINE";
```

the values PRINTed are 24 24, because the semi-colon prevents skipping to the next line.

The following rather artificial program PRINTs a new character over the top of the last character PRINTed:

```
10  PRINT AT 20,0; "ENTER
    VALUES FOR ""PRINT AT"""
20  INPUT A
30  INPUT B
40  PRINT AT 20,0; "ENTER A
    CHARACTER TO PRINT"
50  INPUT Z$
60  PRINT AT A,B;Z$;
70  LET B = 32 - PEEK 16441
80  LET A = 24 - PEEK 16442
90  GOTO 40
```

Colour queries

ONE familiar request which interests many people is: *"Can I use ZX-80/1 programs on my Spectrum?"*

There is no straighforward answer, because of a number of important differences between the three machines. Broadly speaking, a ZX-81 program which does not use PEEK or POKE and contains no machine code routines will run on the Spectrum. Similarly, a Spectrum program which does not use PEEK or POKE, machine code routines, colour, graphics and the other new facilities will run on the ZX-81. Note, however, that cassettes SAVEd on one machine will not load on the others.

Converting a ZX-80 program can be difficult, because it uses integer arithmetic, whereas the ZX-81 and Spectrum use real arithmetic.

"Can you explain how the display file works? I have tried PEEKing and POKEing it but I still cannot understand it".

Look at pages 164 and 165 of the *ZX Spectrum Basic Programming* manual. You will see from the memory map that the display file is at the bottom of RAM between 16384 and 22527 inclusive, with the so-called attributes area at 22528 to 23295. A quick calculation shows that there are 6,144 bytes in the display file and as there 6,144 bytes in the display file and as there are 32 characters in each of 24 lines displayed on the screen, that means that there are eight bytes per character. You can see how those bytes are used by running this program:

```
9000 FOR i = 16384 TO 22527
9010 POKE i; 255
9020 NEXT i
9030 PAUSE 0
```

The screen will be covered gradually with black horizontal lines. Notice that each line is separated by eight vertical steps from its predecessor, that lines are drawn in groups of eight, and that at the end of each group the next line is drawn back at the beginning of the group. There are three such groups.

In effect, the display is in three

separate units and within each unit the first 256 bytes determine the condition of the top one-eighth of each character position. The next 256 bytes determine the next one-eighth of each character position, and so on.

The attributes area is 768 bytes long, i.e., one byte per character position. It is scanned in the logical fashion so that to POKE the fifth character on the second line of the display, for example, you POKE 22527 = 32 = 5. The attribute byte specifies, among other things, the foreground and background colour, so you can only obtain, at most, two colours per character position.

Hence you cannot expect to obtain high-resolution graphics in multiple colours. This routine demonstrates the full range of facilities. It takes some time to produce the display but it is worth it in the end:

```
9000 FOR i = 16384 TO 20480
   STEP 2048
9010 FOR j = i TO i + 2047
9020 POKE j,7
9030 NEXT j
9040 NEXT i
9050 FOR i = 22528 to 23295
   STEP 256
9060 FOE j = 0 TO 255
9070 POKE i + j,j
9080 NEXT j
9090 NEXT i
```

If you are impatient, replace the first five lines by

```
9000 FOR i = 10 TO 704
9010 PRINT "□":REM
   CHARACTER NUMBER 138
9020 NEXT i
```

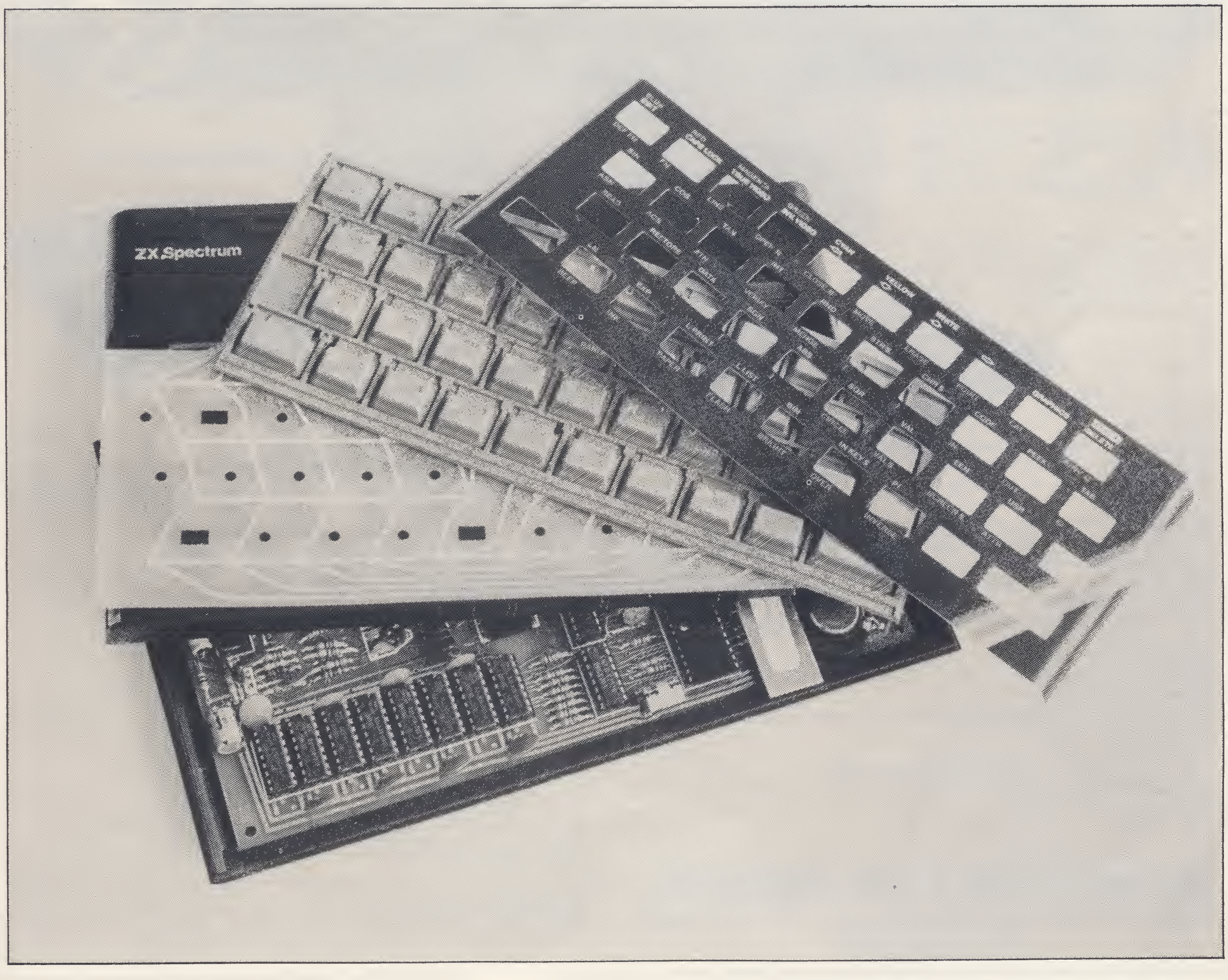
for the same effect but with the bottom two lines missing.

Other potential Spectrum purchasers are wondering whether to buy the 16K or the 48K machines.

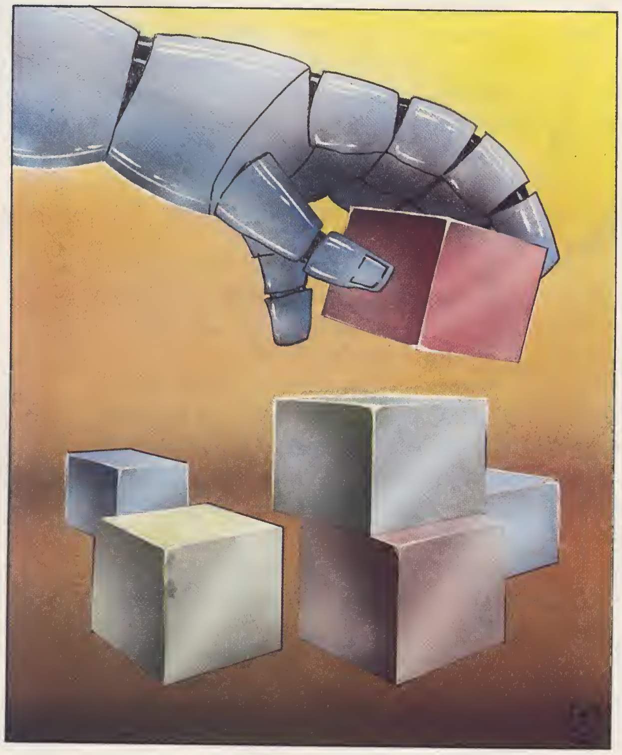
"I have read reports that the 16K Spectrum uses 7K to provide colour and graphics, leaving only 9K of usable memory. There are some marvellous 16K ZX-81 adventure games. Am I correct in thinking that they will not fit into what is left of the standard 16K?"

Broadly speaking, that is correct. A program which just fits into the 16K ZX-81 is unlikely to fit into the Spectrum without a major modification. Many so-called 16K programs, however, do not use all the available RAM. There is also a number of ways in which Spectrum Basic is superior to ZX-81 Basic, so it may be possible to reduce the space required without discarding any of the functions of a program.

For example, the Spectrum allows multiple program lines with each program line, separated by a colon. A colon occupies only one byte in the program area whereas a program line has an 'overhead' of five bytes — one byte for the NEWLINE character, two bytes to hold the line number and two to hold the length of the program line. There is no limit to the number of lines which may be concatenated and so space can be saved. The only thing to stop you putting all your program on one line is that GOTOs and GOSUBs must be made to lines starting with a line number and that IFs skip to the next line number if the appropriate condition is not true.



MACHINE CODE



As programmers become more adept in the use of Basic, they find the language limits their abilities to do everything they wish to do. Using machine code helps to overcome some of the problems. Earlier in the year Mike Biddell wrote a series on how to begin using machine code. We have grouped the articles together and print them as a single article.

Moving faster by machine talk

WHY MAKE the effort to learn machine code? Three main reasons spring to mind. It is faster in operation than the equivalent Basic program. It occupies much less space in the memory than the equivalent Basic program — that is critical on the unexpanded ZX-81. It gives greater scope for the imagination than Basic.

With Basic you are tied to the menu of instructions the manufacturer has given you in the brain — ROM — of the computer. With machine code, you can make the operating system do things outside the set menu, make the screen scroll printed information downwards — the Sinclair menu gives only a scroll up.

So machine code can be packed tighter, run faster and produce more imaginative effects outside the manufacturer's original Basic instruction set.

Machine code is a set of numeric instructions, called bytes, which, when addressed to the Z-80 chip central processing unit, make it perform some sensible function. The Z-80 would not understand a Basic instruction, such as PRINT, directly. A very complex piece of machine code called the Basic Interpreter breaks PRINT into numeric instructions for the Z-80 chip to produce the required effect from the command PRINT.

Those hexadecimal codes — number base 16 for convenience — are listed under Z-80 Assembler in the ZX-81 Manual — pages 181 to 187.

It seems a little daunting but really the fundamental things you can tell the Z-80 chip to do are really simple but the overall effect can be very powerful.

Here is a summary of the main things you will ask the Z-80 to perform during a machine code program; you ask it simply by feeding it the appropriate code:

Add one to the contents of a register — called increment. A register is somewhere in the chip where you can hold a number/value to do something with it.

Take one away from the register — decrement.

Save the contents of the register elsewhere, so you can use that register for something else — Push.

Bring back the saved value into the register — Pop.

Carry-out a machine code routine somewhere else, then return to the same point in the program — Call and Return.

Jump backwards or forwards a number of instructions if a register or part of a register is not zero — Jump Relative No Zero.

Jump backwards on forwards a number of instructions if a register or part of a register is zero — Jump Relative Zero.

Load registers with specific numbers.

They do not appear to be the building bricks of computing power, but they are.

To put machine code into a ZX-81, write a single-byte instruction to an address in the memory. The POKE instruction takes the general form: POKE 16514, 12, where 16514 is the address and 12 the instruction code.

The POKE instruction operates on decimal numbers, so that all the hexadecimal codes must first be converted to decimal before POKEing them into the computer memory. The decimal code 12 instructs the Z-80 chip to increment — add one — to its 'C' register.

The most suitable place for the code, it is generally agreed, is within a REM statement placed at line 1 of the program. The machine code loader — figure one — POKEs your machine code into spaces after the REM statement — addresses 16514 onwards.

So sit at your ZX-81 keyboard, switch on and type it in. There are approximately 100 letter Ms after the REM statement, to reserve space for machine code.

The purpose of line 5 is to list the program, including the REM statement, after each decimal code entry, so that you can see the code going into the REM statement. The Ms in the REM change to the graphic character corresponding to the number entered, or as a question mark if one does not exist. To show this in action, enter the machine code loader again and RUN it.

The program has run but all you will see happen is that the cursor changes from the K to the L mode. Some people can be confused by this and assume that the program has not run, or the system has crashed.

The reason the program was designed this way is to allow the REM statement to be viewed as it changes with each code entered.

It is then possible to view the loader in action by entering the Sinclair graphic codes on page 181 of the Sinclair Manual. After each code i.e., NEWLINE, 1 NEWLINE, 2 NEWLINE, you will see the appropriate characters appear, in turn, in the REM statement. It is not meaningful machine code, of course — it just demonstrates the loader in action.

Figure 1: Machine code loader.

```
   1 REM MMMMMMMMMMMMMMMMMMMMMMM
MMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMM
MMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMM
MMMMMMMM
   3 LET T=16514
   4 CLS
   5 LIST 1
  20 INPUT C
  30 POKE T,C
  40 LET T=T+1
  50 GOTO 4
```

Figure 2:

```
   1 REM ■■TAN MMMMMMMMMMMMMMMMM
MMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMM
MMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMM
MMMMMMMMMMM
   3 LET T=16514
  10 LET A=USR (T)
  20 PRINT A
```

To break out of the loader enter MM NEWLINE. You can then use POKE immediate. You can poke immediate only if the cursor is in the K mode. Try it by entering:

POKE 16514,128 NEWLINE

Do not use a line number and you will see, when you get back the listing, that a large black square has appeared in the first position after the REM statement.

The reason is that although 128 is the Z-80 op-code for adding the contents of the A and B registers in the Z-80 chip, it is also the ZX-81 character code for a black square.

At that point, you can play some interesting tricks using POKE immediate. Try this one:

POKE 16510,0

You will notice that the line number of the first line has changed from a one to a nought. Then enter O NEWLINE to try to delete it and, surprisingly, it will not delete — a useful way of ensuring that the valuable machine code in your REM statement is not deleted accidentally.

Using POKE immediate it is also possible to make the listing disappear from the screen altogether, apart from "0 REM", as follows:

POKE 16514,118
POKE 16515,118

That tricks the ROM into believing the display file has terminated. The machine code loader is, of course, still in the program file and will still run. List 3 will confirm this. Then RUN.

Then we can enter our first machine code program but previously we call a machine code program, using the USR function and the Basic statement as follows:

e.g. 10 LET A = USR (16514)

That tells the program flow to jump to and execute the machine code routine, starting at address 16514. An additional piece of information which can often be used to advantage is that the value given to A is the same as that contained in the B and C registers, combined, of the Z-80 chip.

To get back from machine code routine you must end the routine with the code for RETURN — i.e., 201. Enter the following simple program, using the loader:

Decimal	Nemonic	Comment
1		Load the BC register
1	LD BC, 01	pair with the Value 1
0		
201	RET	Return to the Basic programme

Run the machine code loader then enter 1 Newline, 1 Newline, 0 Newline, followed by 201 Newline. You will then see in the REM statement two small black squares, followed by a space, followed by TAN — all is then well.

Then delete lines 4, 5, 20, 30, 40 and 50 from the program and add lines 3,10 and 20 — figure two.

If you then run the program shown in figure two, a figure 1 will appear on the screen. That shows that your first machine code program has run correctly. You have loaded the BC register pair with the value 1, the program has returned from machine code to Basic successfully and printed the value of A, which is the value contained in the B and C registers as a pair.

The machine code loader can then be used to develop a more complex machine code routine, which will be a useful addition to your subroutine library.

At that stage attempt to write a machine code routine which, when called once, moves every line on the screen down one, leaving the top line blank and making the bottom line disappear. That could have any number of applications either in games or in a rudimentary word processor.

You will need the 16K RAM pack added at this stage. When the screen is full, which is the case with most games, the 1K RAM is

Figure 3: Machine code scroll down for 16K ZX-81.

STEP	HEX CODE	DECIMAL CODE	NEMONIC	COMMENT
1	2A 0C 40	42 12 64	LD HL (40 0C)	Load the display file. Start address into HL.
2	11 72 02	17 114 2	LD DE, 626 DEC	Size of screen to be scrolled.
3	19	25	ADD HL, DE	Point HL at last character on screen to be scrolled.
4	E5	229	PUSH HL	Temporarily store this address on the stack.
5	06 21	6 33	LD B, 33 DEC	Load B register with V.D.U. line length.
6	23	35	INC HL	Point HL to one line below by incrementing.
7	10 FD	16 253	DJNZ – 1	HL 33 times.
8	E5	229	PUSH HL	Temporarily store this address on stack.
9	D1	209	POP DE	Put the HL value off the stack into the DE register.
10	E1	225	POP HL	Bring back the original HL value into HL
11	0E 13	14 19	LDC, 19 DEC	No of lines to scroll.
12	06 21	6 33	LD B 33 DEC	Length of line including line end marker.
13	7E	126	LD A, (HL)	Load A, with the character code pointed to by the HL pair.
14	12	18	LD(DE), A	Load the position pointed to by the DE pair with the character code in A.
15	1B	27	DEC DE	Point DE at the next position.
16	2B	43	DEC HL	Point HL at the next character to be copied down.
17	10 FA	16 250	DJNZ – 4	Repeat above four steps thirty three times (one line).
18	0D	13	DEC C	Reduce line count by 1.
19	20 F5	32 245	JRNZ – 7	Jump back to step 12 if line count not zero.
20	C9	201	RET	Return to BASIC programme.

generally insufficient and the addition of the 16K RAM pack reserves space automatically for a full display. With a collapsed display file, the coding of our scroll down program becomes difficult to understand.

The 1K machine operates with no memory reserved for display and builds-up the display file as you print characters on the screen.

computer it keeps us informed constantly of where it has gone, by putting its new start address (two bytes — sixteen bits) in system variables 16396 and 16397. This is presented as the least significant bit at 16396 (lowest part of the number, i.e., up to 256) and the most significant bit second at 16397 (product of 256).

The number stored at 16397 must

Figure 4: Scroll down test program.

```
    1 REM E£RND)?▀; FAST ▄▀57( CLE
AR FAST SGN LPRINT :(▄▀5$4 PRINT
TAN MMMMMMMMMMMMMMMMMMMMMMMMMMMM
MMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMM
M
    3 LET T=16514
   10 PRINT AT 11,12;"HELLO"
   20 FOR I=0 TO 50
   30 NEXT I
   40 LET A=USR (T)
```

To start writing a machine code routine we first need some knowledge of the registers available in the Z-80 chip. A register is a place/device which can hold one instruction or piece of data (byte) where we can work on it. It is stored in the register as a series of eight zeros or ones (bits) in any combination, for example, 00001100 is the bit pattern for increment (add one to) the 'C' register. For that reason, the Z-80 chip is known as an eight-bit processor.

The registers we will mainly be using are the accumulator (A), H,L, D,E,B,C, and the flag (F) register. The accumulator and flag registers are special-purpose registers; the other six are very similar and can be interchanged and used as pairs such as HL, DE, BC.

They can be used to hold addresses pointing to various parts of the computer memory, because you need 16 bits to address any meaningful quantity of memory.

The accumulator works like any one of the six general-purpose registers but can also be used to perform arithmetical and logical jobs, whereas the general-purpose registers cannot.

The flag register generally is used to indicate whether the result of an operation is zero or not. Testing one bit in this register will tell us, for instance, if subtracting one from the C register resulted in zero. This is used frequently and is very useful.

In the ZX-81, the system shifts the area of memory used for the display; but being a very courteous

therefore be multiplied by 256 to give the higher order part of the address, so to locate the D file address we must evaluate:

PEEK (16396) + 256*PEEK (16397).

At any time, we can locate the start address of the D file by using no line number and PRINT followed by the foregoing expression.

We must often think how we will achieve scroll down in general terms, forgetting about coding for

Figure 5: Main program loop.

```
    1 REM E£RND)?▀; FAST ▄▀57( CLE
AR FAST SGN LPRINT :(▄▀5$4 PRINT
TAN MMMMMMMMMMMMMMMMMMMMMMMMMMMM
MMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMM
M
    3 LET T=16514
    7 PRINT AT 20,9;"**RUNWAY**"
   10 PRINT AT 1,10;"▄▄    .    ▄▄"
   20 LET A=USR (T)
   50 GOTO 10
```

the moment. We shall proceed now writing in words how we expect the program to flow, although we may have to alter our concept later.

Begin by finding the address of the start of the display file and store it in a register pair and then find the address of the end of the display file; as we wish to scroll do not also store it in a register pair, HL registers say. Load the DE register pair with the address directly below HL on the screen; this will involve adding decimal 33 to the HL address, since there are 33 addresses per screen line.

Next shift the character at the position ointed to by HL to the position pointed to by DE. (Repeat for one whole screen line). That will print the whole of the line scanned by HL to the positions below scanned by DE. Shift the addresses of HL and DE up the screen by one line and repeat.

Then repeat until the whole screen has been processed in this way. Each line on the screen has then been copied to the line below so return to Basic (RET).

The completed routine is shown in figure three. It was produced by using the rough list flowchart and the Mostek Z-80 programming manual. It uses the simple programming elements described earlier.

To understand fully the step from flowchart list to the finished program, you should study figure three and the comments in depth, until you can understand to your satisfaction what is happening.

Some confusion can arise because step 13 of the scroll down program (Ld A, (HL), decimal 126) has no equivalent graphics code in the REM statement — not even a question mark.

The reason is that code 126 is used in the program file to tell the ROM that a number is terminated and as such cannot be given visual significance, so the ROM has been instructed not to translate code 126 into the display file.

You need then to enter the code into your computer to test that it works correctly. Ensure that you have your 16K RAM attached or it will not work.

Enter the machine code loader. RUN it and type-in the decimal code, entering NEWLINE after each number.

After entering the last code, enter MM NEWLINE to break out of the program. Then delete all lines except one and three and add lines 10,20,30 and 40 as shown in figure four, the scroll down test program.

When you RUN the program, "HELLO" should be printed in the centre of the screen and, after a short delay, it should move down one

line. Save this program on tape, as later we will explain how the routine might be called from a Basic program to produce an interesting game.

A logical procedure to adopt is to decide the game appearance, strategy and rules. That must be considered in the context of scroll down, since we intend to use the routine we have developed.

Because the scroll down would produce an excellent moving roadway or airfield effect, a game can be built around that idea. Here is a concept of the game plot.

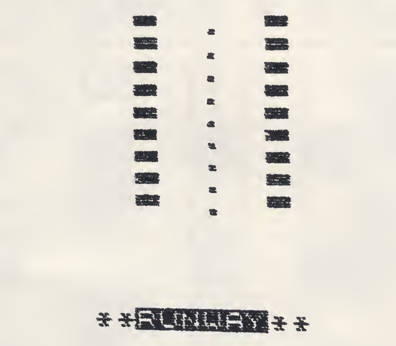
It is war-time and you are flying a transporter aircraft to land supplies behind enemy lines. As the runway approaches, you can see that it is mined but it is too late at that stage to pull up, since your airspeed is too low. You have, therefore, to land and taxi around the mines to take off again and see the runway recede.

If you hit a mine or go off the side of the runway, the aircraft crashes. now produce a series of statements about how the program might flow and operate, as follows:

Main program loop, using PRINT AT function to produce the runway element. Call USR to scroll that element downwards. Put the USR call in a loop and that should produce a moving runway effect.

Add the aircraft path by adding a POKE statement into the loop. Allow the aircraft to be steered left and right by using INKEY $ to modify the POKE address. Scroll down should create a pilot of the previous positions of the aircraft.

Figure 6: VDU picture created by main program loop.



Within the loop, POKE the mines on to the runway, in varying positions. Add the logic for a mine or verge collision. That should send the program pointer outside the loop for a comment and game re-start.

Add a program to advance the taxying speed of the aircraft as it proceeds. Finally, incorporate program lines to signify a win; create a receding runway and a jump outside this loop to comment on the win and allow game re-start.

Now let us make a start with the main program loop. Delete lines 10, 20, 30 and 40 of the scroll down test program and add lines 7,10,20 and 50 as shown in figure five. Line 7 prints the game title on line 20 of the screen; that is not scrolled, since the routine scrolls only 19 lines. Lines 10,20 and 50 produce the approaching runway effect — study this three-line loop carefully.

Then run the program and you should see, if all is well, the runway approach. Press BREAK before the runway reaches the bottom of the screen and you will have a VDU picture similar to that shown in figure six. Our main program loop incorporating the machine code works.

Then add the aircraft path and some means of steering it.

An asterisk (*CHR$(23)) is chosen to represent the position of the craft at any time. Since we will be POKEing the asterisk into the display file we will have to locate it, as described previously, by PEEKing system variables 16396 and 16397 and using those to define a variable W.

Then add the following lines to the developing program:

4 LET V = 411
5 LET W = PEEK 16396 + 256*PEEK 16397
30 LET V = V + (INKEY $ = "8")—(INKEY S = "5")
40 POKE W + V, 23

Variable W in line 5 defines the start address of the display file and variable V in line 4 is the displacement to be added, to position the asterisk in the correct place on the screen.

The POKE statement in 40 places the asterisk on the screen, directly below the runway. The value of V was determined by trial and error.

The steering logic is in line 30. It uses INKEY $ to decode keys 5 and 8 on the keyboard.

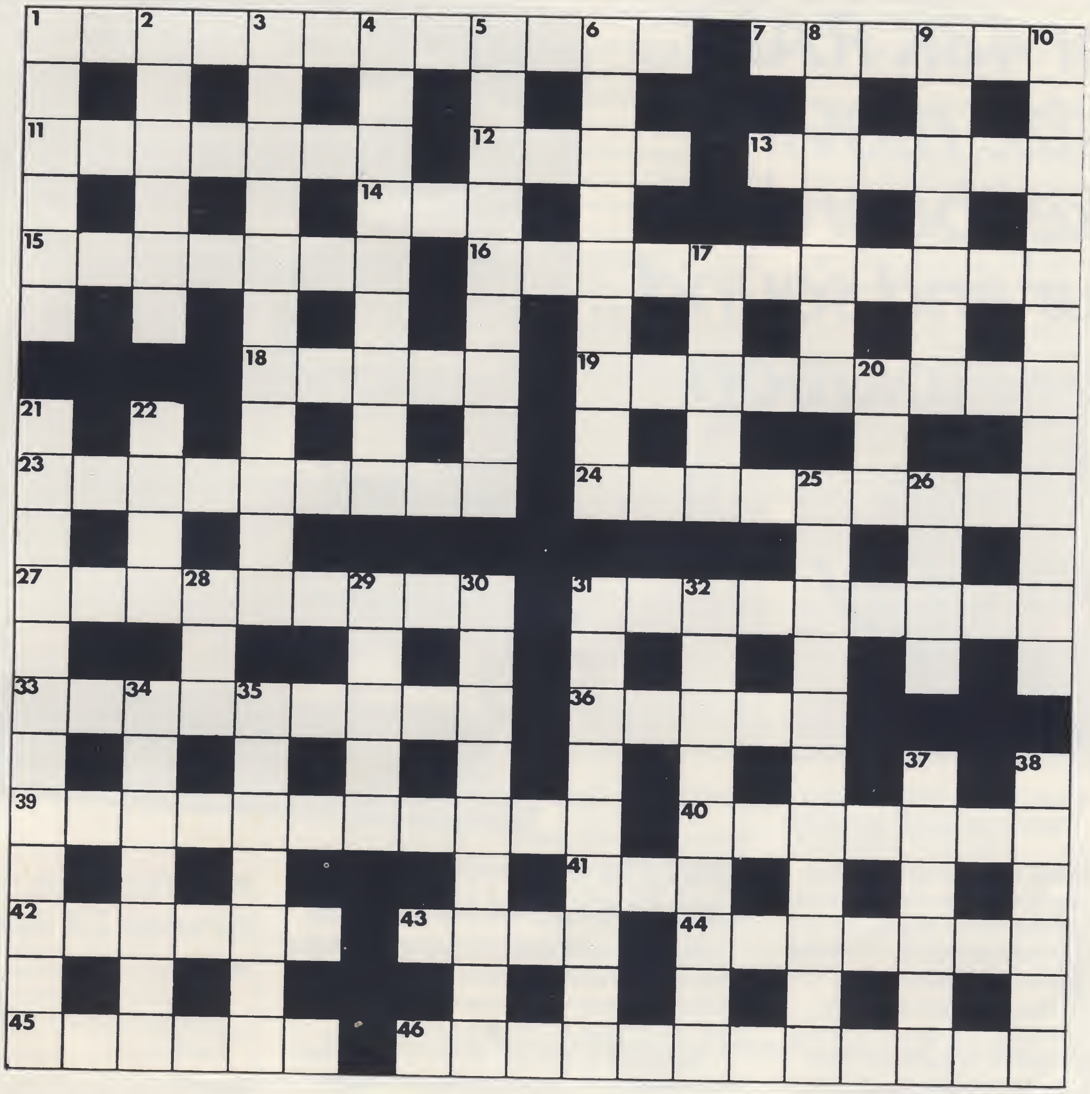
We have covered items one and two of the main program specification. It would be excellent practice to add your own lines.

For those who feel less inclined to tackle the task, a completed program — it is only one solution — is presented in figure seven.

Figure 7: 'Runway' — complete program listing, Basic and machine code.

```
   1 REM E£RND)?"; FAST "57( CLE
AR FAST SGN LPRINT :("5$4 PRINT
TAN MMMMMMMMMMMMMMMMMMMMMMMMMMMM
MMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMM
M
   2 LET S=0
   3 LET T=16514
   4 LET V=411
   5 LET W=PEEK 16396+256*PEEK 1
6397
   6 CLS
   7 PRINT AT 20,9;"**RUNWAY**"
  10 PRINT AT 1,10;"■    .    ■"
  20 LET A=USR (T)
  21 LET R=INT (S/10)
  22 IF S-R*10=9 OR S-R*10=6 OR
S-R*10=4 THEN POKE W+V-33*INT (V
/33)+99,52
  30 LET V=V+(INKEY$="8")-(INKEY
$="5")
  35 IF PEEK (W+V)=3 OR PEEK (W+
V)=52 THEN GOTO 100
  40 POKE W+V,23
  41 IF S=100 THEN LET V=V-66
  42 IF S=150 THEN LET V=V-33
  43 IF S=200 THEN LET V=V-99
  44 IF S=250 THEN GOTO 250
  45 LET S=S+1
  50 GOTO 10
 100 POKE W+V,61
 110 PRINT AT 21,0;"TOO BAD YOUR
DEAD REPLAY Y/N?"
 120 INPUT Y$
 130 IF Y$="Y" THEN RUN
 140 GOTO 120
 250 PRINT AT 1,10;"           "
 260 LET V=V+(INKEY$="8")-(INKEY
$="5")
 270 IF PEEK (W+V)=3 OR PEEK (W+
V)=52 THEN GOTO 100
 280 POKE W+V,23
 290 LET S=S+1
 300 IF S=270 THEN GOTO 400
 310 LET A=USR (T)
 320 GOTO 250
 400 PRINT AT 21,0;"WELL DONE RE
PLAY Y/N?"
 410 INPUT X$
 420 IF X$="Y" THEN RUN
 430 GOTO 410
```

CROSSWORD



Across
1. You must be one as it's before you now (8,4)
7. Bray in pain in computer code . . . (6)
11. . . . as it's to do with numbers as crime unfolds (7)
12. U.S. standard-bearer 64)
13. How data travels (2,4)
14. Diary of a mathematical function (3)
15. State the period to mean (7)
16. Not the op-code in a machine instruction (7,4)
18. Glint on a computer case (5)
19. Zero output (2,7)
23. What to do with a root — to make beer? (7,2)
24. Trickiest hardware fault to find (4,6)
27. Thus sayeth the Sinclair guide (1,2,6)
31. Goes in for processing (5,4)
33. Excitement about the effect of RANDOMIZE? (9)
36. Chopper helps paper tape to keep going (5)
39. Coding instructions (11)
40. Return the data to health? (7)
41. Car overturned by U.S. hardware manufacturer (3)
42. Aid nun to reach African chief (6)
43. American soldier comes to Georgia with a billion (4)
44. Most fit me this way before the bench-mark (7)
45. Where all roads go? (2,4)
46. Accept the need for an empty loop to 25 (5,2,5)

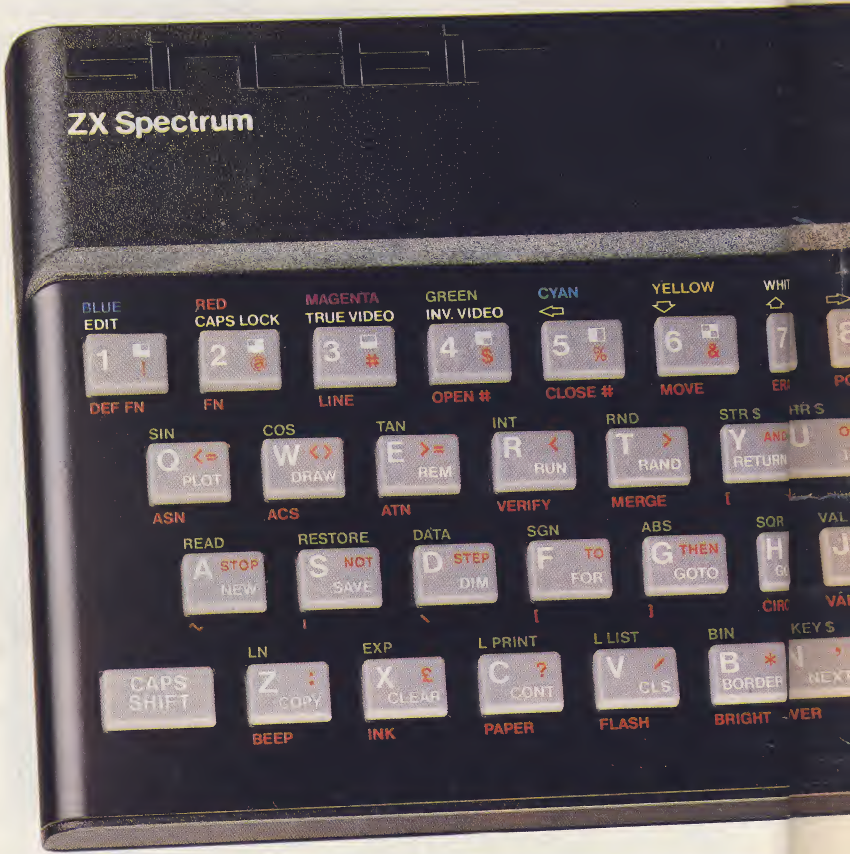
Down
1. Crime-rate, I hear, for Basic structure (6)
2. Less-sensitive numeric constant (6)
3. Situation of dormant bugs? (6,5)
4. Lent n mice the change for awful weather (9)
5. — the insurmountable barrier (2,7)
6. Assumed end of a Sinclair IF . . . THEN statement (4,2,3)
8. Sin sits around but won't take "no" for an answer (7)
9. She takes a large beer, I hear (7)
10. Reply of syntax checker to a valid statement? (3,2,2,5)
17. What a report code often reports (5)
20. Ferranti's chip (3)
21. Tiny sign of non-integer arithmetic (7,5)
22. Part of a PRINT statement (4)
25. Slow down to make the program more useful (3,3,5)
26. Overcome the CPU time unit (4)
28. Describes a running Sinclair (3)
29. Source of Spectrum and of Spectrum software (5)
30. Carrying-out benchmark tests without error? (3-6)
31. Fit the whole numbers together? (4)
32. Computer buff's answer to a problem (7,2)
34. To base on ASCII maybe . . . (2,5)
35. . . . or RAM, ROM and 12 (7)
37. To stagger and spool (2,4)
38. With impudence deal with a critical path? (6)

Answers on page 114

Sinclair ZX Spectru

16K or 48K RAM... full-size moving-key keyboard... colour and sound... high-resolution graphics...

From only £125!



First, there was the world-beating Sinclair ZX80. The first personal computer for under £100.

Then, the ZX81. With up to 16K RAM available, and the ZX Printer. Giving more power and more flexibility. Together, they've sold over 500,000 so far, to make Sinclair world leaders in personal computing. And the ZX81 remains the ideal low-cost introduction to computing.

Now there's the ZX Spectrum! With up to 48K of RAM. A full-size moving-key keyboard. Vivid colour and sound. High-resolution graphics. And a low price that's unrivalled.

Professional power– personal computer price!

The ZX Spectrum incorporates all the proven features of the ZX81. But its new 16K BASIC ROM dramatically increases your computing power.

You have access to a range of 8 colours for foreground, background and border, together with a sound generator and high-resolution graphics.

You have the facility to support separate data files.

You have a choice of storage capacities (governed by the amount of RAM). 16K of RAM (which you can uprate later to 48K of RAM) or a massive 48K of RAM.

Yet the price of the Spectrum 16K is an amazing £125! Even the popular 48K version costs only £175!

You may decide to begin with the 16K version. If so, you can still return it later for an upgrade. The cost? Around £60.

Ready to use today, easy to expand tomorrow

Your ZX Spectrum comes with a mains adaptor and all the necessary leads to connect to most cassette recorders and TVs (colour or black and white).

Employing Sinclair BASIC (now used in over 500,000 computers worldwide) the ZX Spectrum comes complete with two manuals which together represent a detailed course in BASIC programming. Whether you're a beginner or a competent programmer, you'll find them both of immense help. Depending on your computer experience, you'll quickly be moving into the colourful world of ZX Spectrum professional-level computing.

There's no need to stop there. The ZX Printer – available now – is fully compatible with the ZX Spectrum. And later this year there will be Microdrives for massive amounts of extra on-line storage, plus an RS232 / network interface board.

Key features of the Sinclair ZX Spectrum

- Full colour – 8 colours each for foreground, background and border, plus flashing and brightness-intensity control.
- Sound – BEEP command with variable pitch and duration.
- Massive RAM – 16K or 48K.
- Full-size moving-key keyboard – all keys at normal typewriter pitch, with repeat facility on each key.
- High-resolution – 256 dots horizontally x 192 vertically, each individually addressable for true high-resolution graphics.
- ASCII character set – with upper- and lower-case characters.
- Teletext-compatible – user software can generate 40 characters per line or other settings.
- High speed LOAD & SAVE – 16K in 100 seconds via cassette, with VERIFY & MERGE for programs and separate data files.
- Sinclair 16K extended BASIC – incorporating unique 'one-touch' keyword entry, syntax check, and report codes.

Z
ca

no
Pr
inc
ho
Inv
His
...
for
of
gra
de

R
in

ye
ZX
te

as
po
sy
RC

Si
Ca
Te

*SPECTRUM MONITOR
*SPECTRUM EDITOR/ ASSEMBLER
*ZX81 SCREEN KIT 1
*ZX81 ZX-MC *ZX81 REMLOAD

FAST MAIL ORDER-SEND TODAY!

SPECTRUM MONITOR. Machine Code Debug/Disassembler
• Enter, Run, Debug machine code programs. • Compatible with Basic • Breakpoints and Registers Display • Disassembly to screen and/or ZX Printer • 16K and 48K versions on one cassette + 30 page manual. **£7.50**

SPECTRUM EDITOR/ASSEMBLER. A powerful and essential machine code programming aid. Major features include:
16K & 48K on same cassette with full documentation.
EDITOR with Auto Line Numbering: 40 Column screen display, tabulated into fields for easy reading: 5 character Label Names: simple Line Editing and Cursor Control: SAVE/LOAD Text Buffer to cassette: output to ZX PRINTER.
TWO-PASS ASSEMBLER accepts all Z80 mnemonics (plus many unpublished mnemonics): Decimal or Hex numbers: simple arithmetic on operands: Assembler Directives — ORG, END, DEFB, DEFW, DEFS, DEFL, EQU, DEFM.
WE CANNOT FULLY DESCRIBE THIS IMPORTANT UTILITY HERE, AND ASK YOU TO SEND A S.A.E. FOR COMPLETE DETAILS OF THIS AND ALL OUR PROGRAMS. **£8.50**

ZX81 SCREEN KIT 1. More power to your screen in all your Basic programs
BORDERS any size, anywhere on screen. SCROLL in all 4 directions. CLEAR and REVERSE PART OF SCREEN. FLASHING CURSOR anywhere on screen — simulates INPUT. DATA FILES SAVE AND LOAD Basic variables: Double Speed. 880 bytes machine code for INSTANT RESPONSE. Becomes part of Basic Program. **£5.70** 4K to 64K

ZX81 ZX-MC. Machine Code Debug/Monitor
• ENTER, RUN, DEBUG machine code programs • SAVE, LOAD, VERIFY at double speed • BREAKPOINTS and REGISTERS DISPLAY • Self contained — cannot be used with Basic • Cassette plus 36 page manual. **£7.50** 4K to 64K

ZX81 REMLOAD. Machine Code Debug/Monitor
• A version of ZX-MC without the Save/Load/Verify facility • Compatible with Basic • CREATE A REM LINE of any length • BREAKPOINTS and REGISTERS DISPLAY • Cassette plus 30 page manual **£6.95** 16K to 64K

**SEND S.A.E. NOW FOR DETAILS
6 Corkscrew Hill, West Wickham, Kent BR4 9BB.**

Prices include VAT & P&P

PICTURESQUE
PICTURESQUE
PICTURESQUE
PICTURESQUE

ZX81-Spectrum
MANAGEMENT GAMES

AIRLINE — £5
Can you compete with British Airways? You must decide on the number of aircraft to operate, whether to buy or charter, and the levels of staffing and maintenance. Problems encountered are tax demands, strikes, cancelled flights, hijacks and aircraft crashes.

AUTOCHEF — £5
As MD of a Catering Company you must negotiate for leases, decide on menu prices, level of wages, advertising and dividends. Each year you must predict the inflation rate. You will be made to resign if you are not successful. There are 3 levels of difficulty.

PRINT SHOP — £5
You own a small printing company and are required to decide on (a) the number and type of staff you employ and when to increase or reduce staff (b) the amount and type of paper you stock (c) the week in which work is scheduled (d) the quotation for each. There are 3 scales of difficulty.

FINANCIAL MODELLER
MODELLER X — £8
This is a user friendly business modeller which provides Managers with a tool to plan the strategic decisions of a business. The model shows the profit sensitivity and the interaction of the pertinent marketing and production factors of a business. A user manual is included in the price of the program.

DISCOUNT £1.50 for two, £3 for three and £5 for four.
ALL PROGRAMS INCLUDE DETAILED INSTRUCTIONS AND NEED **16K.**
EXTENDED 48K SPECTRUM VERSIONS AVAILABLE AT £1 EXTRA PER PROGRAM.
PLEASE STATE COMPUTER TYPE AND SEND CHEQUE TO:

C.C.S.
**DEPT. SA
CASES COMPUTER SIMULATIONS
14 Langton Way
London SE3 7TL.**

SILVERSOFT
ZX81/Spectrum Computer Software

'ORBITER' & 'GROUND ATTACK' Simply the most amazing ZX-Spectrum arcade games available.

ORBITER Fast and furious action is what you get in this amazing Defender-style program for the ZX-Spectrum.

All the excitement of the arcade inhabits your micro. Aggressive aliens flash across the screen as you try to destroy them with your laser-bolts and smart bombs, while rescuing the humanoids.

Orbiter is written entirely in m/c code and has full arcade features, including scanners, reverse, hyper-space, continuous scoring and sound effects, plus humanoids, landers, mutants and all the other alien nasties which make it one of the best games around.

GROUND ATTACK Survival is the name of the game in this exciting Scramble-type arcade game for the ZX-Spectrum.

Your mission is to pilot your spaceship through tortureous caverns while destroying the enemy missile launchers and fuel dumps. Endurance as well as quick thinking are needed to survive as the caves become progressively smaller and airborne aliens start to appear. No-one has yet survived!!

GROUND ATTACK is written completely in machine code. It has 26 zones, full: up, down, forward and reverse, controls. Lasers, bombs, explosions, continuous scoring and sound effects, plus rockets, fuel dumps and airborne aliens.

Both games will run in either 16K or 48K Spectrum and cost £5.95 each.

Programmers NOW you can make money from your micro. Send your programs to us today. We pay probably the best royalties around.

GENEROUS DEALER DISCOUNTS AVAILABLE
Silversoft Ltd, 20 Orange Street, LONDON WC2H 7ED.

NEW! ZX81-COMPILER Yes! Now you can write machine code on your ZX81. No more messing about with assemblers and disassemblers simply type in the BASIC program and the machine does the rest. You'll never need to buy another machine code program again!! ONLY £5.95.

ZX81 ARCADE ACTION

MUNCHER	
–Exciting pacman game for the ZX81	£4.95
ASTEROIDS	
–"Just the thing for asteroid addicts" PCW Sept 82	£4.95
INVADERS	
–"Probably the best version of INVADERS you will get for the ZX81" – PCW '82	£3.95
ALIEN-DROPOUT	
–Exciting ORIGINAL arcade game for your ZX81	£3.95
STARTREK	
–YES! you can be a starship commander	£3.95
GRAPHIC GOLF	
–18 graphically displayed holes to test even the best golfer	£3.95
SUPERWUMPUS	
–An enthralling underground adventure for your ZX81	£3.95
GAMES PACK	
–Fantastic value for money, nearly 50K of programs on one cassette! Only	£3.95

Please send me ______
I enclose a cheque/PO for £ ______
Name ______
Address ______

Silversoft Ltd, 20 Orange Street, LONDON WC2H 7ED. SU

Answers to

CROSSWORD

ACROSS:
1, SINCLAIR USER;
7, BINARY; 11, NUMERIC;
12, ANSI; 13, AS BITS;
14, LOG; 15, AVERAGE;
16, ADDRESS PART;
18, SHEEN; 19, NO RESULTS; 23, EXTRACT IT;
24, TORN CABLE; 27, I AM HELPER; 31, INPUT DATA;
33, AGITATION; 36, TOOTH;
39, PROGRAMMING;
40, RESTORE; 41, RCA;
42, INDUNA; 43, GIGA;
44, MEETEST; 45, TO ROME;
46, AGREE TO DELAY.

DOWN:
1, SYNTAX; 2, NUMBER;
3, LARVAL STAGE;
4, INCLEMENT; 5, UP AGAINST; 6, ELSE DO NOT;
8, INSISTS; 9, ABIGAIL;
10, YES IT IS LEGAL;
17, ERROR; 20, ULA;
21, DECIMAL POINT;
22, ITEM; 25, CUT THE SPEED; 26, BEAT; 28, HOT;
29, PRISM; 30, RUN-TIMING;
31, INTEGRATE:
32, PROGRAM IT; 34, IN ORDER; 35, ACRONYM;
37, TO REEL; 38; PERTLY.

"QS Produce the most ULTRA-MEGA-AMAZING Games in the entire known Universe!"

TIMEGATE

Quicksilva's 1st level two game. The best space/time adventure of 82.

TIMEGATE
High speed graphics combine with electrifying space adventure makes TIMEGATE the game of all games.
Foil the ruling aliens by travelling back through time and blasting out their home planet. Fight 3D space battles on your journey, seeking the time gates that lead to year zero and the salvation of the human race.
Features: 4D, High speed graphics, versatile scanning, status and battle computers, land & take-off from many planets, time-travel, 5 skill levels, score & rating plus free training program.

MORE NEW RELEASES!

THE CHESS PLAYER

Who is he? Dare YOU challenge him? Risk high stakes and pit your wits against the chess game with personality, speech and devastating chess skills. THE CHESS PLAYER costs less, while offering you more skill—boosting enjoyment.
Features: 6 skill levels, analysis, copies display and all moves to printer or screen, colour or B & W display, save game at any point, Personality & speech.

SPEAKEASY

Add speech or music to you programs! In response to massive demand, we have released SPEAKEASY for your own use and enjoyment.
Simply record your sound onto a cassette and play it into your Spectrum via load. The signal may then be edited and replayed on command from your Basic or M/c program.
NO EXTRA HARDWARE REQUIRED!

Edward de Bono's Classic
L-GAME

Simple rules and complex strategies make the L-GAME an essential for everyone.
Also includes outstanding and compulsive versions of Mastermind & Pontoon.
As featured on T.V. in Edward de Bono's Thinking Course.
L-GAME REACHES THE BITS OTHER GAMES DON'T TOUCH!

SPACE INTRUDERS
Features 4 alien types, random saucer points, 2nd attack wave mutant, exploding aliens, bonus base 50,000 points, left-right & fire, sound effects and colour.

METEOR STORM
3 meteor types, 2 saucers, full screen wrap-around, bonus ship every 10,000 points, amazing explosions, left-right-thrust-fire & hyperspace, sound effects and speech.

WHAT THEY SAY ABOUT OUR ZX81 GAMES:

QS SCRAMBLE—
"...amazing, fantastic!..." [P.C.W.]

QS INVADERS—
"...just like the real thing!..." [C & V.G.]

QS ASTEROIDS—
"...very good." "...addictive game..." [C & V.G.]

QS DEFENDER—
"...Better than any other arcade game I've seen..." [Sync]

"...nicest games I've played on the ZX81. All the tapes were impressively packaged..." [Which Micro & S.R.]

COME AND SEE OUR AMAZING RANGE AT:
THE NORTHERN COMPUTER FAIR [25th - 27th Nov]
THE NEXT ZX MICROFAIR [Mid December]

ZX - 81 GAMES

Scramble	m/c 16k £3.95		☐
Invaders	m/c 16k £3.95		☐
Asteroids	m/c 16k £3.95		☐
Defenders	m/c 16k £3.95		☐

SPECTRUM GAMES

Speakeasy	m/c 48k £4.95		☐
Timegate	m/c 48k £6.95	*	☐
The Chess Player	m/c 48k £6.95		☐
Space Intruders	m/c 16k £4.95	*	☐
Meteor Storm	m/c 16k £4.95	*	☐
L-Game	Basic 16k £3.95		☐

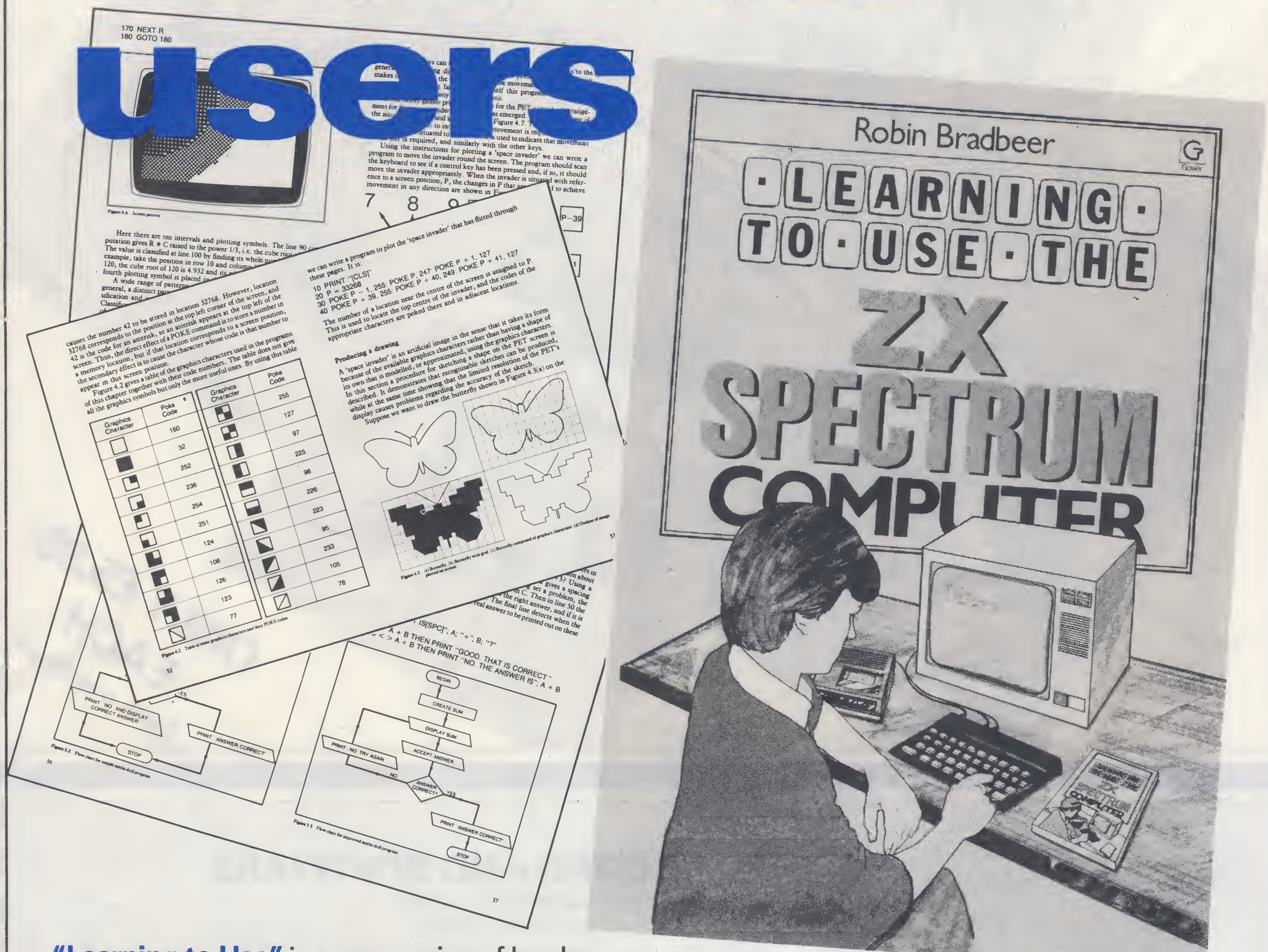
***THESE GAMES INCLUDE SPECIAL SEASONAL DISCOUNTS, ORDER NOW AND GET FREE KEYBOARD OVERLAY!**

Please send me the games as ticked.
Total cheque/P.O. enclosed ....
Name ....
Address ....
....
....

Send order to: QUICKSILVA, DEPT SPA, 92 NORTHAM ROAD, SOUTHAMPTON, SO2 OPB

JUST AVAILABLE

A new series for first time users



"Learning to Use" is a new series of books which introduce newcomers to the most widely used micros in the marketplace.

The books assume absolutely no knowledge about computers and the reader is shown even the most fundamental operations such as "switching on" and "loading a program". The books lead the reader through simple programming and then on to graphics, with several programs which show how to achieve pictures and even animation!

The user-friendly approach is consistent throughout the text – not only are program listings clearly shown, but in many cases, a photograph is included to show what the program looks like when actually loaded and run! All books in the series are £5.95 (incl. postage).

Gower.

Order form overleaf.